DEFENCE R&D DÉFENSE

Development of Control and Synchronization of Periodicity and Chaos in Recurrent Neural Networks for Robotic Applications

Z. Tan and M.K. Ali
University of Lethbridge

Scientific Authority:
S.A. Barton
Defence Research Establishment Suffield

WARNING

"The use of this information is permitted subject to recognition of proprietary and patent rights."

DEFENCE RESEARCH ESTABLISHMENT SUFFIELD

Contract Report
DRES CR 1999-110

July 1999

Défense nationale National Defence

Canada

102-02300/

Development of Control and Synchronization of Periodity and Chaos in Recurrent Neural Networks for Robotic Applications

Z. Tan and M. K. Ali

Department of Physics
University of Lethbridge

Contents

Introduction

Report 1

Report 2

Report 3

Report 4

Report 5

Introduction

In this treatise, we present the results of our work that has been funded by Defence Research Establishment Suffield (DRES) since 1997 under R&D contract W7702-6-R607/A and the project title "Development of Control and Synchronization of Periodicity and Chaos in Recurrent Neural Networks (RNNs) for Robotic Applications". Standard RNNs involve nonchaotic dynamical systems with fixed points that serve for storage and retrieval of memories. This approach severely limits the scope of applications of RNNs simply because nonchaotic RNNs constitute a small fraction of all possible RNNs. We embarked on treating RNNs from a much broader perspective and in the light of recent developments in dynamics. This required studying chaotic as well as nonchaotic RNNs. A brief description of our findings is given in this introduction.

Pattern recognition by chaotic neural networks is a big challenge because there is no persistence in the dynamics. This property determines the essential difference between a chaotic network and a nonchaotic one. It means that the pattern recognizing strategies prevailing in nonchaotic systems are no more valid in chaotic systems. From the viewpoint of dynamics, nonchaotic networks used to recognize patterns have attractors in the phase space. These attractors are fixed points and usually stable. Therefore, they are designed to be connected to the memories or patterns to be retrieved. One of such networks is the Hopfield model, a model that has played a vital role in developing artificial neural networks. In this model, the attractors are defined to be the patterns. The information on the patterns are encoded in the synapses connecting the cells. Memory retrieval is realized in the dynamical process, when the system is started from a state near a pattern (attractor). However, we cannot utilize this procedure in chaotic systems. The reason is that a chaotic system does not have stable fixed points - every point in the state space can be reached from every other point through dynamical evolution. Furthermore, chaotic systems are sensitive to initial conditions, which implies that chaotic systems lose memory rapidly.

Why should we study chaotic neural networks? The answer is that they are unavoidable or even necessary in real biological systems. Experiments in neuroscience suggest that chaos influences the behavior of nervous system.

In their work, for example, Freeman and Skarda show that chaos helps rabbits recognize odors, by switching them from one odor to another. Even in the classical Hudgkin-Huxley work, investigators have found the necessity to introduce chaos analysis. In addition, the tendency of applying the chaos theory to clinical cases has been receiving growing interest. Contributions from various areas show the importance and necessity of chaos in neural systems.

Since 1997, we have been interested in artificial neural networks with chaos. We concentrated on associative memory that is utilized by many intelligent systems including the human brain. For pattern recognition by chaotic networks, we have introduced a number of approaches which we tried on various systems. These approaches include chaos control, virtual basins of attraction (root finder), full synchronization and phase synchronization. We have studied the memory retrieval in the Gardner model, the Aihara model, the chaotic Rössler/Lorenz system and coupled lattices. Our results suggest that associative memory can be demonstrated in systems with chaos. The details of our studies are attached as **Reports** numbering from 1 to 5. These reports were submitted to Dr. Simon Barton during the progress of the work

Report 1 deals with a model used by Dr. Barton. The dynamics can be convergent (ordered) and chaotic depending on the parameters involved in the model. In the convergent case, this model can be used to recognize patterns. In the chaotic case, trajectories normally do not converge to fixed points. By introducing relative hamming-distance and chaos control, we have shown that the existence of fixed points are not necessary and hence this model is able to recognize patterns, when the dynamics is chaotic or non-chaotic. The recognition has the character of short-term memory, in both the learning and recognizing processes. The synchronization of the chaos control guarantees a small distance between the pattern and the state. The time of synchronization can be utilized to determine the pattern which the state should be recognized to be.

Report 2 treats the Gardner model by using chaos control. We hope to find a general approach to recognize patterns. This approach does not need stable fixed points which do not exit in the chaotic model. The chaotic model stores information in the synapses for a short time. Using feedback

to synchronize two trajectories representing the patterns and states to be recognized, we introduce the time of synchronization as our criterion. The shortest time gives us the information about patterns.

Report 3 has reviewed the associative memory in various models such as Aihara *et al*'s model, Tsuda's model, Nara *et al*'s model, the Gardner model and others. In addition to discussing the approaches used in the individual models, we have introduced the method of virtual basins based on algorithms, such as the Newton-Raphson method, for finding roots (fixed points) of dynamical systems. This method is applicable to chaotic as well as nonchaotic systems. Here, we have also used synchronization for pattern recognition. The root finder method is applied successfully to a chaotic model, which provides us a way to determine the dynamic persistence that is associated to the pattern recognition and increase greatly the storage load. Various version of synchronization has been demonstrated in diverse discrete and continuous coupled lattices such as cubic map, logistic map and continuous Hopfield model. As stated above, stable fixed points are not required for pattern recognition.

Report 4 uses distance synchronization to deals with Aihara *et al*'s model and introduces phase synchronization in the system of coupled Lorenz oscillators. Aihara *et al*'s model has been shown to give the best result when it is applied to Travel Sales Man Problem. In this report we address the associative memory problem that has not been solved in an acceptable way. Recent experiments suggest that the phase synchronization would play important role in brain activities. Our results show that phase synchronization is applicable to recognition in coupled Lorenz systems, typical chaotic systems.

Report 5 is about stochastic resonance (SR) in intelligent systems, a novel direction in chaos related studies. Recent experiments in neuroscience show the phenomenon of stochastic resonance in a neural network from mammalian brain. It becomes to be generally concerned that if this remarkable phenomenon exists in all intelligent systems whose information processing (IP) depends on associative memory (AM), an important mechanism in biosystems. Using a generic model that is able to describe all characteristics of AM in diverse systems, we have revealed that SR exists in a very universal sense and it plays a vital role in IP by neurosystems.

In summary, we have demonstrated that chaotic neural networks can be used to recognize patterns. In terms of the techniques used in nonchaotic networks, the lack of dynamic persistence is the most difficult problem in pattern recognition using chaotic models. However, by devising new approaches such as the time of full and phase synchronizations and virtual basins of attraction, we have shown that chaotic systems are more versatile than their nonchaotic counterparts with regard to pattern recognition. For example, chaotic systems have much larger storage capacity than the traditional nonchaotic ones. The direction of stochastic resonance and information processing by neurosystems is drawing interdisciplinary attention. Our work strongly suggests universal existence of stochastic resonance in the regime of information processing. More work needs to be done in all these areas.

The authors thank Dr. Simon Barton for his continuous guidance, in-depth discussions on the main issues and help at all stages during the progress of the project. Financial support from DRES is thankfully acknowledged. The authors also thank Brian Hepburn and Chris Tucker for allowing to use the part of the work in which they were co-investigators.

Report 1

Chaotic Reccurent Neural Network

(July - September, 1997)

Contents

1 Introduction **3**

2 Pattern Recognition with Chaotic Dynamics **5**
2.1 The effect of time in learning process 8
2.2 The dynamic hamming-distance in recognition 11

3 The Short-Term Memory in Recognizing Process **16**
3.1 The effect of time in recognizing process 16
3.2 The synchronization of chaos in pattern recognition 21

4 Conclusions **24**

Chapter 1

Introduction

Recently, Simon Barton has used an interesting neural network model [1, 2, 3] whose dynamics can be convergent (ordered) and chaotic depending on the parameters involved in the model. In the convergent case, this model can be used to recognize patterns, that means it can store patterns and retrieve them, on the basis of short time dynamics and relative hamming-distance in the space of states. Compared with the usual recognition process in models such as the Hopfield model [4, 5, 6] and the Gardner model [8, 7, 9], where patterns are stored in fixed points, the convergence to fixed points is not utilized in Barton's work although fixed points may exist at the end of the trajectories in the ordered case. In the chaotic case, trajectories started with arbitrarily chosen initial points normally do not converge to fixed (periodic) points. One then asks the question: does the model used by Barton fail to recognize patterns in chaotic cases? In this report, we argue that the existence of fixed (periodic) points are not necessary in Barton's work and the model can be used for pattern recognition in convergent as well as chaotic situations in terms of relative hamming-distance.

For pattern recognition in Barton's work, we assume that the information about a pattern is embedded in the history of the trajectory and our task is to exploit it. It may be mentioned that in Barton's work, all trajectories in the convergent case eventually end up at the same fixed point. Hence, arriving at the fixed point will not help in pattern recognition in this model. Here, short time dynamics before the fixed point is reached, is relevant for pattern recognition. The recognition mechanism is dependent on the time

for learning and the time for recognizing and there exist optimum values for learning and recognizing times. It has in fact the character of "short-term memory". Based on these considerations, we show that the model under consideration can still recognize patterns in the sense of "relative dynamic hamming-distance". The relative hamming-distance gives us information about the stored pattern, although it does not guarantee an infinitely close convergence of a twisted state to a stored pattern. For example, if we have stored many patterns in the network, the relative hamming-distance tells us to which pattern is the state closest.

In addition to the idea of relative hamming-distance, we also discuss the absolute hamming-distance between the state and the pattern. We introduce a dynamic absolute distance and a procedure of synchronization to measure the recognition. Work in this area is in progress.

In Chapter 1, we discuss first the pattern recognition and short time effect in the original model. We will see that the relative distance is a satisfactory criterion to describe the recognition, if we don't care for the absolute distance. The effect of duration in the *learning* process is shown clearly. Then in Chapter 2 we discuss the behavior of recognition and the synchronization in a modified model. The effect of time in the *recognizing* process is studied. Because in the original model, the sensor nodes in the model do not participate in the dynamics, which yields finite fixed difference between the evolution of patterns and the state to be recognized, it is not convenient for synchronization. In the modified model, the synchronization and related issues can be discussed much more clearly. Conclusions and future aspects are presented in Chapter 4.

Chapter 2

Pattern Recognition with Chaotic Dynamics

The typical picture of recognition is that if a state is started near a pattern in the space of states, after some dynamic steps it will evolve to the pattern stored in the network. The evolution gives a trajectory in the space of states. The distance between the state and the pattern is given by the hamming-distance. In models such as the Hopfield model [4, 5, 6] and the Gardner model [7], the pattern is a fixed point (attractor) under the dynamics. A pattern is recognized by a persistent fixed point to which the dynamics evolves [6, 8, 7, 9]. As mentioned above, the model proposed by Barton can be chaotic or ordered depending on the system parameters. In the ordered region, all states eventually evolve to the same fixed point (see below). A pattern in the Barton's approach is recognized using 'short-term memories' generated by a learning rule. Clearly, the pattern recognition process of this model is different from that of the Hopfield type networks which use fixed points for pattern recognition. We argue that the short time dynamics of Barton's model can accomplish pattern recognition in both ordered and chaotic regions. What is needed is to compute the 'relative dynamic hamming-distance' at every time step as the system evolves.

In Barton's model, there are three kinds of neurons (nodes), the sensor nodes (snodes) accepting the stimulus from outside, the recurrent nodes (rnodes) processing the stimulus (information) from sensor nodes and the motor nodes (mnodes) responding to the stimulus. Every sensor node is connected

to a number of rnodes and every motor node is connected to a number of rnodes, too. Both of these connections (weights) are set to be constant. In addition, every rnode has also connections with other rnodes. The connections among the recurrent nodes, however, obey a learning rule.

Let N_s, N_r and N_m respectively represent the number of sensor nodes, recurrent nodes and motor nodes. One of the nodes, say the i-th one, at time t is connected with c_i (chosen randomly) nodes through weights W_{ij}^t with $j = 1, \ldots, c_i$ (refer to [2] for detail). The output (state) of the i-th node at time t is given by a sigmoid function

$$f_i^t = [1 + \exp(-s(x_i^t - x_{0i}^{t-1}))]^{-1}, \tag{2.1}$$

where, the parameter s and an offset x_{0i}^{t-1} determine the steepness and the position of the sigmoid function, respectively. x_i^t is the total input to the i-th node

$$x_i^t = \sum_{j=1}^{c_i} W_{ij}^{t-1} f_j^{t-1}. \tag{2.2}$$

The learning rule of the weights of the *recurrent* nodes are

$$\Delta W_{ij}^t = \alpha s_{ij} f_i^t f_j^{t-1} - \gamma W_{ij}^t, \tag{2.3}$$

where, s_{ij} is the sign of the weight, parameter α controls the amplitude of the change and γ is a decay parameter. In addition, the offsets of the recurrent nodes change in the following way

$$\Delta x_{0i}^t = \beta(f_i^t - 0.5) \tag{2.4}$$

which makes the output approach the center, where β is a constant. Therefore in the stable state, the outputs of rnodes are 0.5. The model, established in this way, presents stable (convergent), periodic and chaotic behavior, depending on the parameters s, α, β, γ [2]. In the stable case, for example, $s = 0.1, \alpha = 0.005, \beta = 0.002, \gamma = 0.0005$, the network evolves eventually to the fixed point. Actually the output of the i-th recurrent node after some sufficient number of steps is 0.5 because of Eq. (2.4). By the use of criterion of closest hamming-distance [3], this model can work as an associate memory and can recognize patterns. In this report, we are concentrating on the pattern recognition with $s = 1.5, \alpha = 0.01, \beta = 0.02, \gamma = 0.001$ which lead to

a chaotic dynamics.

Because the motor nodes are connected with fixed weights to the recurrent nodes, it is reasonable to study only the states of the sensor nodes and the recurrent nodes which evolve according to the dynamics (2.1). Suppose we have stored P patterns, $\xi^\mu, \mu = 1, \ldots, P$ which are encoded in the weights through a learning process. Then we start the system by giving a state, S, twisted from one of the patterns (the twist is often referred as noise). Independently, the patterns are also input to the system to start the system dynamics. During the dynamics we get one (recognizing) trajectory yielded by the twisted state and P (master) trajectories produced by the P patterns. We add a index, μ, to f_i^t to denote the output of the i-th node at time t for pattern μ as $f_{\mu i}^t$. The distance between the recognizing trajectory and the master trajectory μ is

$$h_\mu(t) = \sum_{i=1}^{N_s}(f_{\mu i}^t - s_i^t)^2 + \sum_{i=1}^{N_r}(f_{\mu i}^t - s_i^t)^2, \tag{2.5}$$

where, $f_{\mu i}^t$ is the state of ξ_μ at time t and s_i^t is the state of S at time t. The first term in Eq. (2.5) is determined by the initial values and remains to be unchanged, because the sensor nodes do not join the dynamic process. We have considered in fact two evolution's, one is the evolution of patterns ξ_μ whose state at time t is described by $f_{\mu i}^t$ and the other is the evolution of the input state S whose state at time t is denoted by s_i^t. $h_\mu(t)$ is thus the dynamic distance at time t. If after some time, $h_\mu(t)$ approaches zero or it gets very small, we believe that the state S is recognized to be the pattern ξ_μ. If $h_\mu(t)$ is not small, we think that the smallest distance

$$h^{min}(t) = min\{h_1(t), \ldots, h_P(t)\} \tag{2.6}$$

tells us that S will be recognized to be the pattern which has $h^{min}(t)$ with S. In fact this criterion has been used to recognize patterns in the convergent case [3]. We discuss below the effect of time of learning (training) in pattern recognition.

2.1 The effect of time in learning process

In the chaotic case, the pattern can not be defined to be an attractor and hence the learning process will not stop, we have to truncate the learning process. A reasonable truncation time is such that it yields a distinct minimum distance, $h^{min}(t)$, with the largest probability. If we start a recognition by giving a state S near one of the patterns, say ξ_1, the reasonable time for truncation is such that the probability of $h_1(t) = h^{min}(t)$ is as large as possible.

Fig. 2.1(a) gives the probability $p(t) \equiv p(h_1(t) = h^{min}(t))$ vs. t for learning time of $t_l = 1, 20, 100, 500$, when the state S is twisted with 10% noise. (note: t defined in Eq. (2.5, 2.6) and in Fig. 2.1(a) is the time for recognition of patterns (recognizing time), t_l is the time to learn patterns (learning time). The effect of recognizing time will be discussed in the next section). Fig. 2.1(a) shows that the shorter the learning process, the larger the possibility with an optimum value of the learning time. When the learning process lasts longer, the chaos becomes also stronger which dominates the system and wash out the memory. The competition between memory and chaos can be controlled by adjusting the time appropriately. Fig. 2.1(b) and Fig. 2.1(c) give the similar relation for 30% and 50% noise, respectively. In Fig. 2.1(b), for example, with $t_l < 20$, the dynamic probability $p(t > 0)$ is larger than the initial probability $p(t = 0)$.

This phenomenon of short-term memory is, however, not so obvious in the ordered region shown in Fig. 2.1(d) giving the probability $p(t)$ for different learning time. Fig. 2.1(d) shows much smaller differences among $p(t)$ for various t_l. It should however be reminded that t_l should be much shorter that the time required to reach the fixed point. This is so because arrival at the *same* fixed point from different initial points means a loss of memory.

Another aspect observed from the figures is that $p(t)$ varies with t, the recognizing time, although it reaches quickly an equilibrium value. This implies that there may be also short term effect in the recognizing process. In fact, for the modified model, the effect becomes much clearer and we study this feature in the next chapter. If we define the probability $p(t)$ to be the rate of recognition, we can judge easily if a pattern can be recognized, or more exactly, with which probability the pattern can be recognized. Fig. 2.2(a), (b), (c) and (d) give the rate versus various noises for $t_l = 1, 20, 100, 500$,

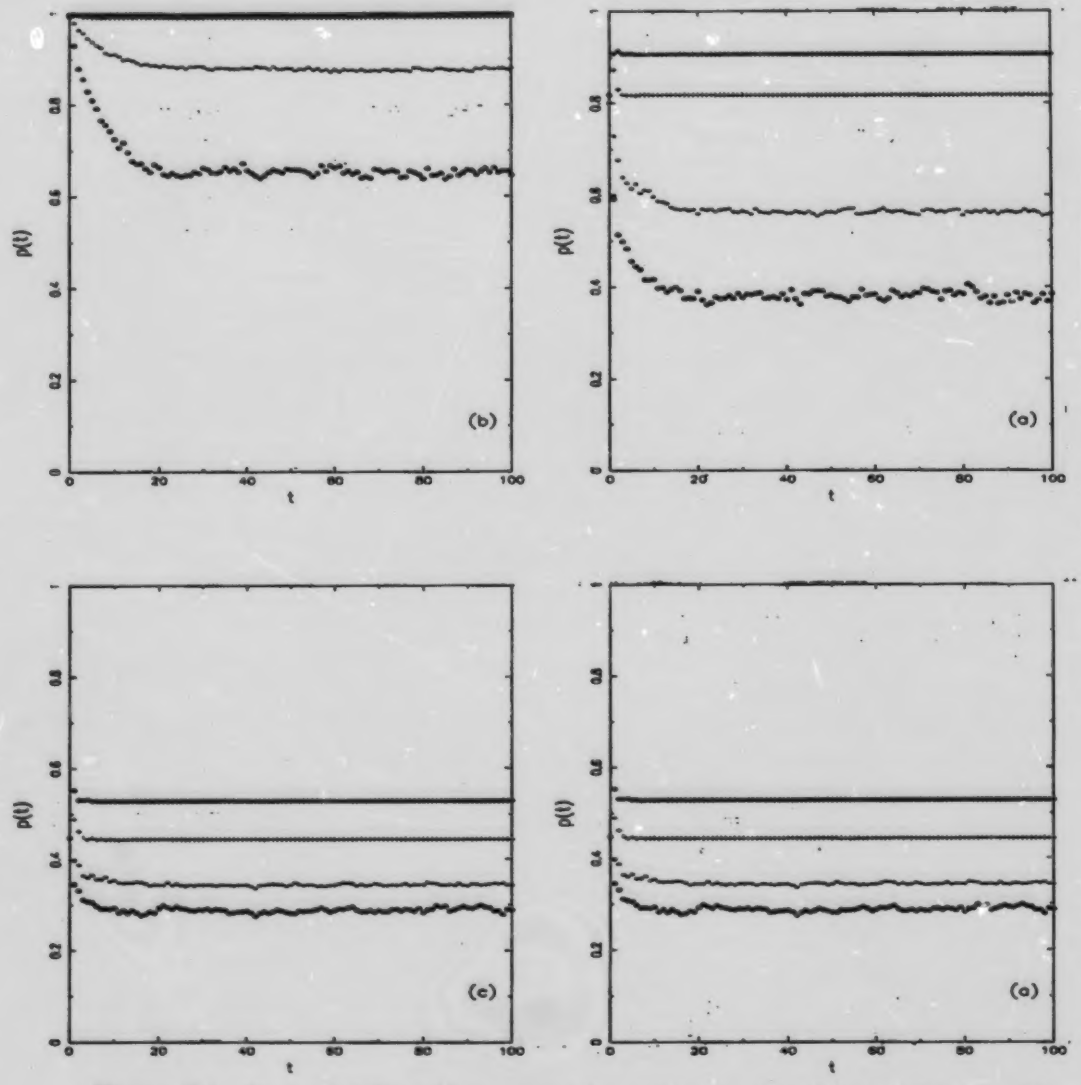

Figure 2.1: The probability $p(t)$ vs. time t for various learning time t_l and noises: (a) noise=0.1, (b) noise=0.3, (c) noise=0.5. (d) noise=0.3 for the case without chaos. The cross-circles, crosses, stars and circles (from top to bottom) represent $t_l = 1, 20, 100, 500$, respectively.

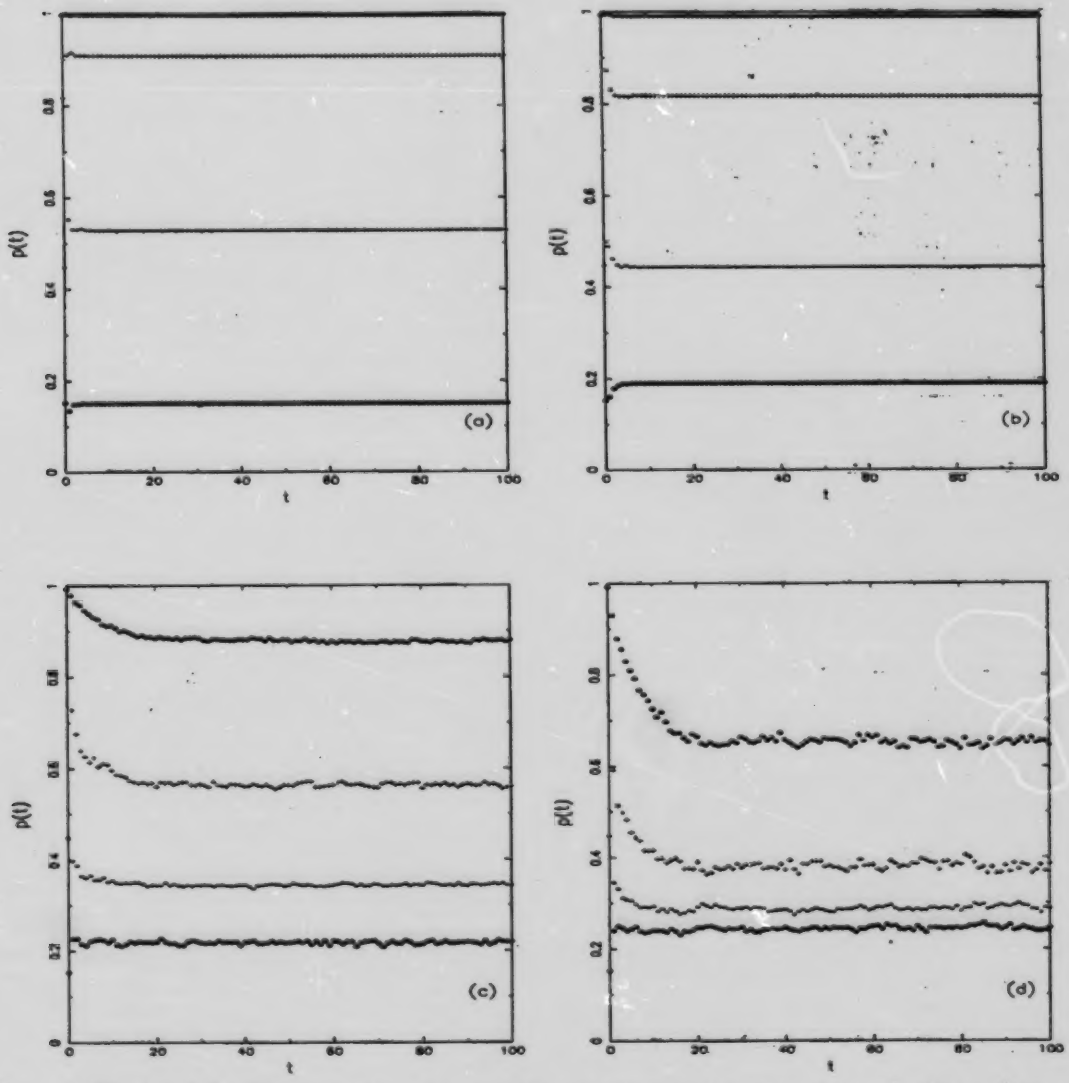

Figure 2.2: The rate of recognition with various noises and t_l: (a) t_l=1, (b) t_l = 20, (c) t_l = 100 and (d) t_l = 500. The cross-circles, crosses, stars and circles (from top to bottom) represent noise 0.1, 0.3, 0.5, 0.7, respectively.

respectively. The equilibrium values are listed in Table 2.1. The rate of

Table 2.1: The rate of recognition (ror)

noise	0.1	0.2	0.3	0.5	0.7
$ror_{t_l=1}$	1.	0.97	0.91	0.53	0.15
$ror_{t_l=20}$	0.99	0.93	0.82	0.44	0.19
$ror_{t_l=100}$	0.88	0.73	0.56	0.34	0.22
$ror_{t_l=500}$	0.65	0.48	0.37	0.29	0.24

recognition (ror) is affected by the noise and the time of learning: for smaller noise (< 0.3) and shorter time, the results are satisfactory.

2.2 The dynamic hamming-distance in recognition

We have discussed the short term memory with the distribution of the relative hamming-distance. "short term" means that we truncate the dynamic learning process and hence avoid the strong chaotic behavior. If we have to deal with a really strong chaotic system generated, for example, by using a longer learning process (t_l is not small) or by twisting the state with larger noise, the criterion used above from the distribution is not effective and we have to turn to another approach as discussed below.

Up to now, we have not discussed the distance defined in Eq. (2.5) which is used in the normal pattern recognition. It is necessary to introduce the average distance as a macroscopical observable quantity, because of the statistical property of the system. This approach has in fact been utilized in the statistical treatment of the Hopfield model [5, 6] and the Gardner model [7, 9]. We define an average distance

$$h_\mu^0(t) = \langle h_\mu(t) \rangle, \tag{2.7}$$

where, $\langle \cdots \rangle$ denotes the average over an ensemble of the system. It can be done easily with numerical treatment. Eq. (2.7) is a dynamic quantity measuring the distance at every time step.

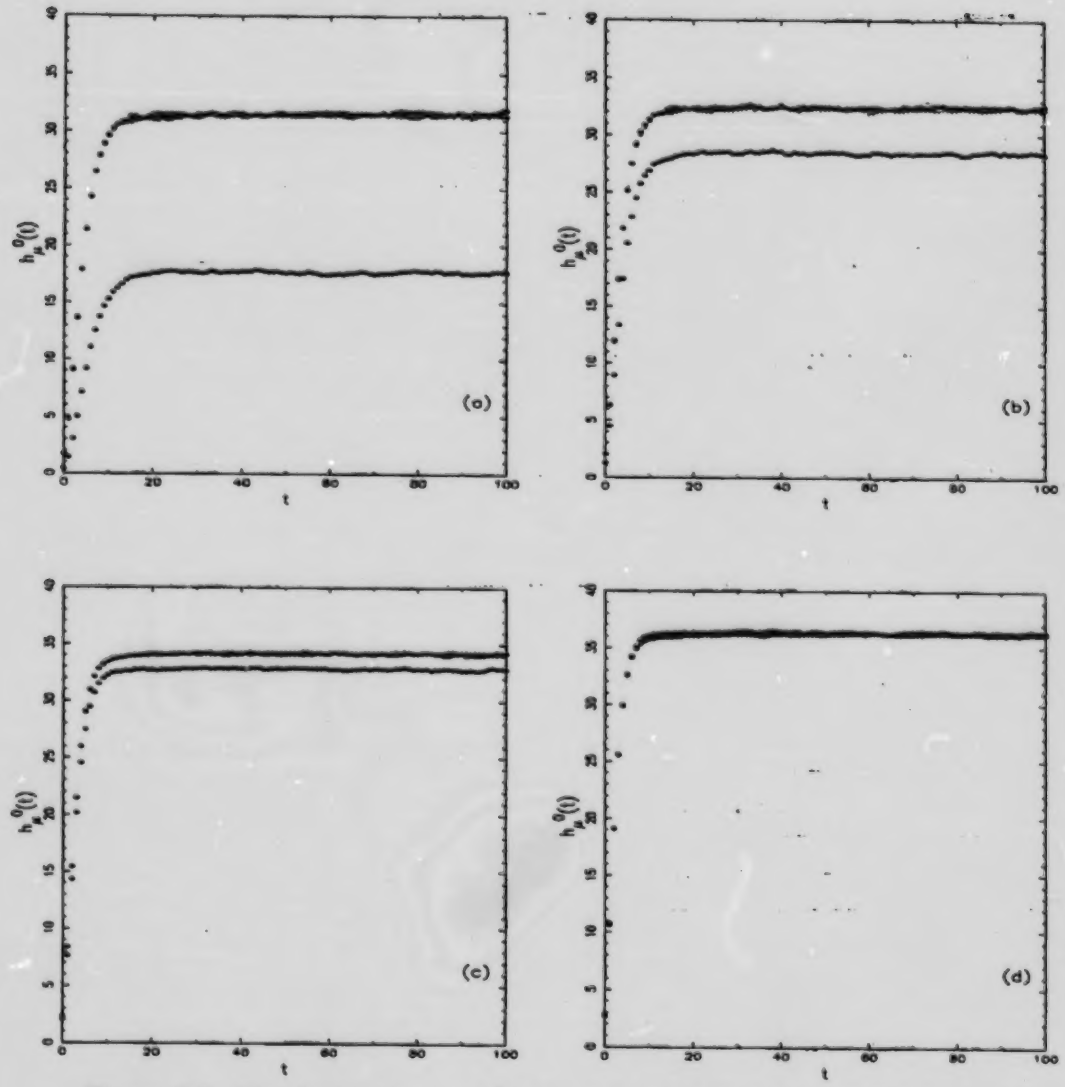

Figure 2.3: The absolute distance $h_\mu^0(t)$ vs. t for $t_l = 500$: (a) noise 0.1, (b) noise 0.3, (c) noise 0.5 and (d) noise 0.7. The cross-circles, crosses, stars and circles represent the distances between S and ξ_1, ξ_2, ξ_3 and ξ_4, respectively. The later three distances are very close to one another and are evidently larger than $h_1^0(t)$ for noise less than 0.7.

Fig. 2.3 gives $h_\mu^0(t)$ versus t, when 4 patterns ($P = 4$) are stored in the network and the state S is given by twisting ξ_1 with noise 0.1, 0.3, 0.5 and 0.7. Fig. 2.3(a), (b) and (c) show that S can be recognized to be closest to ξ_1, because $h_1^0(t)$ is the smallest of $h_1^0(t), h_2^0(t), h_3^0(t)$ and $h_4^0(t)$. From the figure it is seen that $h_2^0(t), h_3^0(t)$ and $h_4^0(t)$ are very closed to one another, because S is not related to ξ_2, ξ_3 and ξ_4. As long as $h_1^0(t)$ is the minimum, S is closest to ξ_1. Combined with the criterion of relative distance used in section 2.2, we could conclude that the system is capable of recognizing S. Therefore, the existence of a minimal $h_\mu^0(t)$ means a successful pattern recognition. The procedure is influenced by the noise which is put to the state S: for small noise, a minimal $h_\mu^0(t)$ exists as shown in Fig. 2.3(a), (b) and (c), while for large noise, the minimal $h_\mu^0(t)$ disappears as in Fig. 2.3(d). But one may ask if the initial distance determines the dynamic distance, i.e. if $h_\mu^0(0)$ is the minimum, then $h_\mu^0(t > 0)$ will be minimum, why we need the dynamics? The answer is that the relation exists only for small noise *after* we have utilized the dynamics. When noise is large, the relation is no more valid. In order to determine the critical value of noise, we have in fact used the dynamics. Fig. 2.3(d) shows that the distances can not be distinguished from one another with noise 0.7. This property will be observed more clearly in a modified model to be discussed in the next chapter, where the existence of the minimal distance is related to the recognizing time. Anyway, we identify the pattern recognition as the existence of a distinct minimal distance.

However, the absolute distance shown in Fig. 2.3 is not small, compared with the same quantity in the convergent case shown in Fig. 2.4, where the typical value is 10^{-2} (which can be identified as recognition, see e.g. the Hopfield model [5, 6]). In the chaotic case, the dynamics drives the state S away from patterns (absolute distances are increasing with the increment of dynamic time), although from the relative distances we can still say that the state is closer to pattern ξ_1. If we insist on that the recognition is identified with a small *absolute* distance, we have to impose an external force to "help" the state S to move to the pattern. This external force should be small (at least for a period of time), because it should not obliterate the original dynamics which propagate the information from the initial situation. Because it is a chaotic system, we turn naturally to the concept of synchronization used widely in chaos control [10, 11, 12, 13]. We utilize a feedback approach

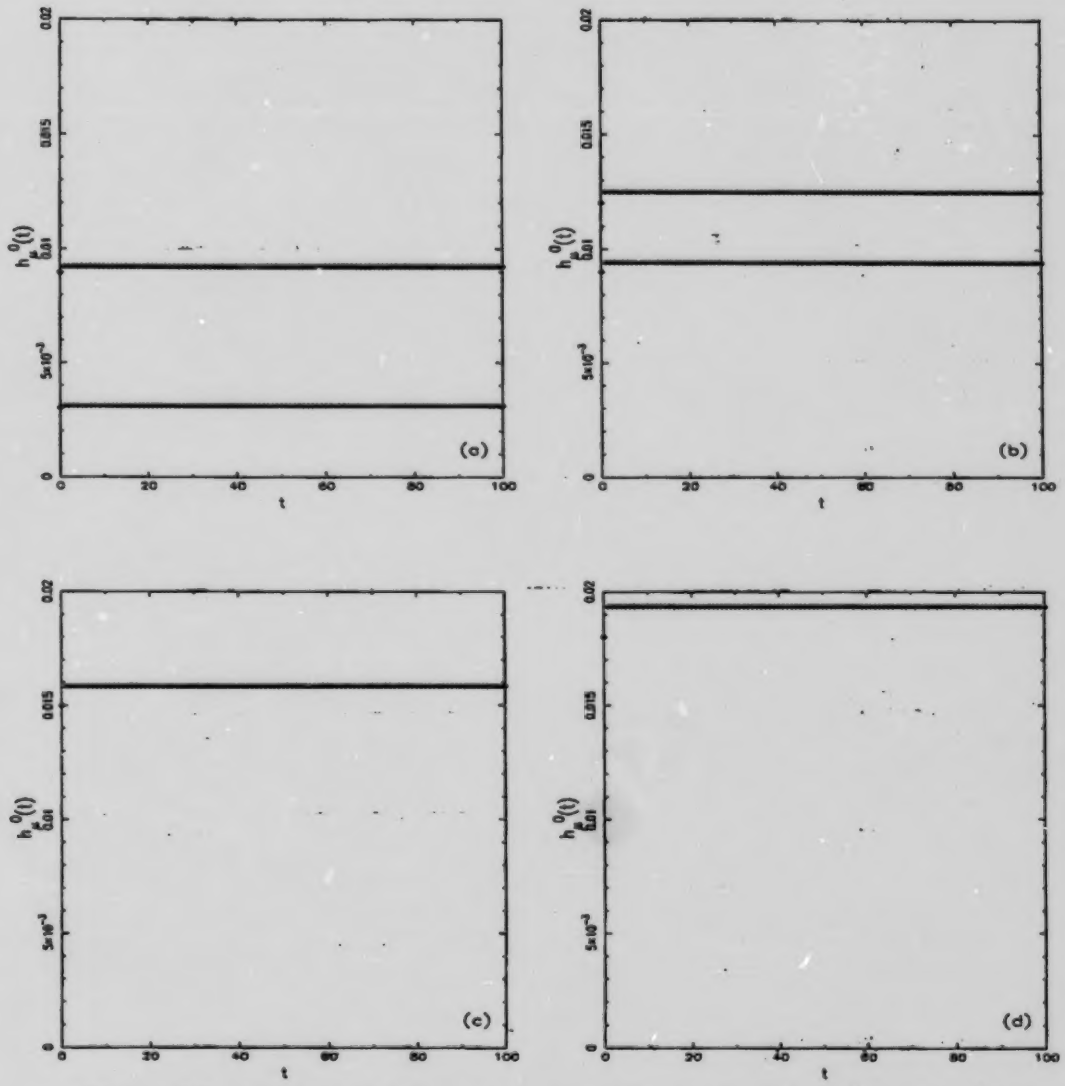

Figure 2.4: The absolute distance $h_\mu^0(t)$ vs. t for $t_l = 500$ in the case where there is no chaos: (a) noise 0.1, (b) noise 0.3, (c) noise 0.5 and (d) noise 0.7. The cross-circles, crosses, stars and circles represent the distances between S and ξ_1, ξ_2, ξ_3 and ξ_4, respectively. The later three distances are very close to one another and are evidently larger than $h_1^0(t)$ for noise being 0.1 and 0.3.

to introduce the external force. After long enough time, the distance between the state and the pattern will be zero. But in this model, the state has always a finite difference with the pattern, because of the structure of the model: the sensor nodes receive the information which is somewhat different from the stored pattern and the sensor nodes do not participate in the dynamics. At every dynamic step, the difference remain to be unchanged. Therefore the absolute distance will not be arbitrary small, even with strong feedback procedure. In the next section, we modify the model and show how the feedback will be introduced there. By the way, the original model is also not convenient to discuss the "short-term memory" in the recognizing process.

Chapter 3

The Short-Term Memory in Recognizing Process

In order to discuss the effect of time spent in the recognizing process and the synchronization with feedback, we need to do some modification of the model. The modification is aimed st introducing the dynamics for the sensor nodes. A slight change can satisfy such requirement. Actually we have modified the model in two ways and obtain two modified models. Here we report one of them. The other will be discussed and reported later. Instead of separating the sensor nodes from the recurrent nodes, we take that the sensor nodes are some of the recurrent nodes. These recurrent nodes chosen randomly accept initially the stimulus from the outside and then they update with the dynamics in the same way as the other recurrent nodes. The same structure is also used in many other models [5, 7]. It means that the sensor nodes are not special nodes. They are a part of the dynamic nodes. Only at $t = 0$, they are given initial values. In this model, we discuss the effect of time in the recognition process and synchronization.

3.1 The effect of time in recognizing process

Fig. 3.1 gives the absolute hamming-distance for $t_l = 1, 20, 80 and 1000$, when 4 patterns are stored and a state S is started near ξ_1 with noise 0.3. First we have observed the effect of learning time discussed in section 2.1, in terms of different behaviors of the hamming distance: for $t_l < 80$, the dynamic

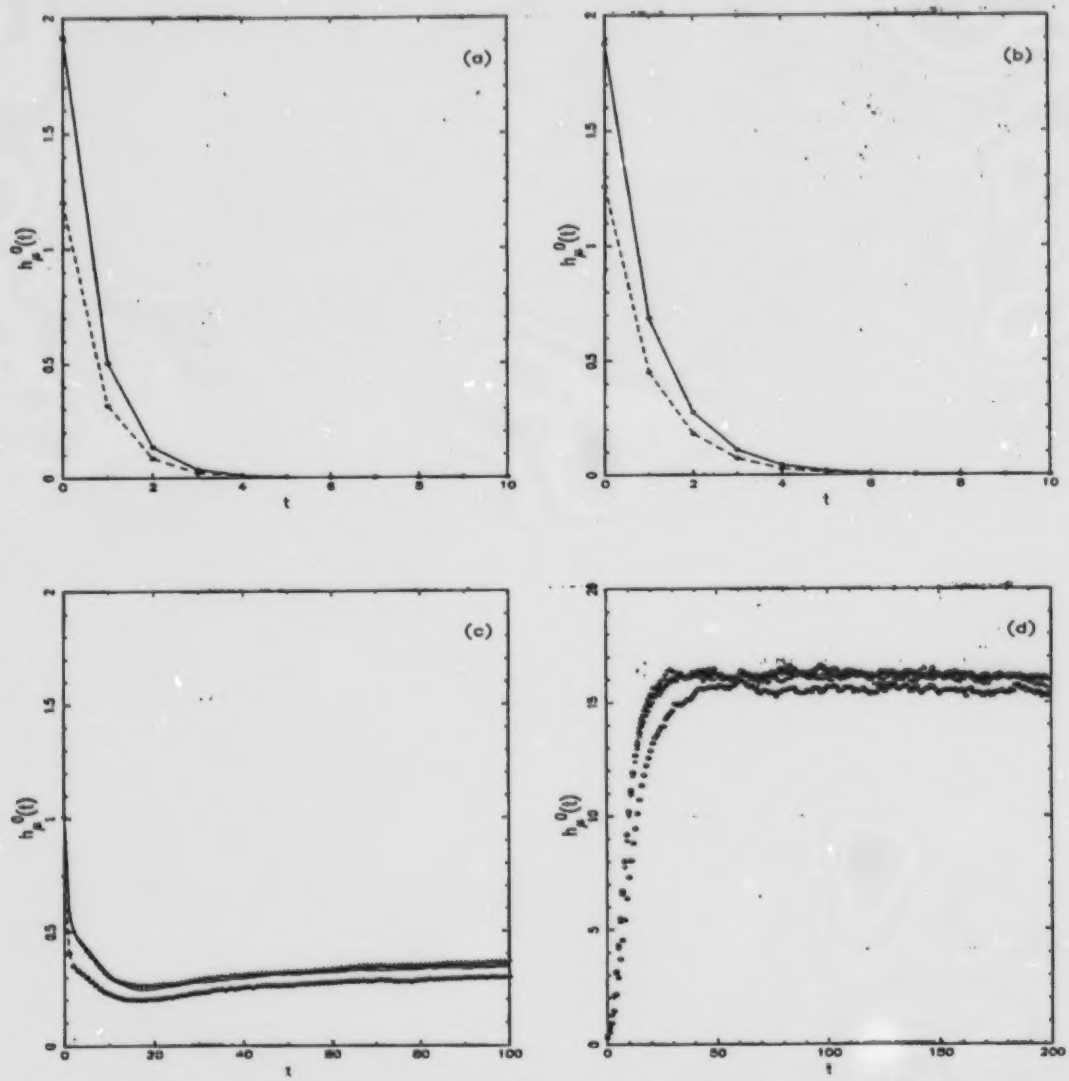

Figure 3.1: The hamming-distance $h_\mu^0(t)$ vs. t for noise 0.3 in the modified model with (a) $t_l = 1$, (b) $t_l = 20$, (c) $t_l = 80$ and (d) $t_l = 1000$. The cross-circles, crosses, stars and circles represent the distance between S and ξ_1, ξ_2, ξ_3 and ξ_4, respectively. The later three distances are very close to one another and are evidently larger than $h_1^0(t)$. In (a) and (b), the points are linked for better perspective.

distance is smaller than the initial distance, $h_\mu(t > 0) < h_\mu(t = 0)$, and $h_\mu(t > 0)$ goes to zero after some steps, while for $t_l > 80$, the dynamic distance is larger than initial distance $h_\mu(t > 0) > h_\mu(t = 0)$. However, it is important to note that, in the figures, there is always a minimum distance. This distance is just the one between the state and the pattern to which the state is initially closest.

Fig. 3.2 gives the absolute hamming-distance for $t_l = 1000$, noise being $0.1, 0.3, 0.5$ and 0.7. It shows that a minimal distance does not exist for ever: it is affected by the noise and the time of recognition. When the noise is larger and the recognizing time is longer, the distance will mix with others and hence the recognition fails. Fig. 3.2(a) and (b) show that the minimal distance exists in the whole range of time, while (c) shows that the distance is minimum for $t < 50$ and (d) tells us such minimal distance does not exist. Smaller noise means that the state has initially more information of the stored pattern. Fig. 3.2 implies that the initial information will be kept for some time depending on the strength of the information. Therefore from this point of view, we could say that *a chaotic system possesses a short-term memory.*

The existence of minimal distance can be shown more clearly if we define an average relative distance

$$H(t) = \frac{1}{P-1} \sum_{\mu=2}^{P} (h_\mu(t) - h_1(t)), \tag{3.1}$$

where, we have assumed that the state is twisted from ξ_1 with noise. Fig. 3.3(a) and (b) show clearly that for noise 0.1 and 0.3, $H(t)$ is larger than zero. For noise 0.5, $H(t)$ is positive at earlier dynamic steps shown in Fig. 3.3(c). If we truncate the recognizing process at a right time step, for example at $t = 50$ with noise being 0.5, the system is able to recognize patterns. This gives the evidence of the short-term memory. When noise is 0.7, $H(t)$ is smaller than zero even at the beginning and it fluctuates around zero afterwards. Hence, the state cannot be recognized for this high level of noise. This is in agreement with the results shown in Fig. 3.2. The height of the hill of $H(t)$ and the speed going to zero are determined by the noise. It means that from the height and the speed, we can obtain the information embedded in the state.

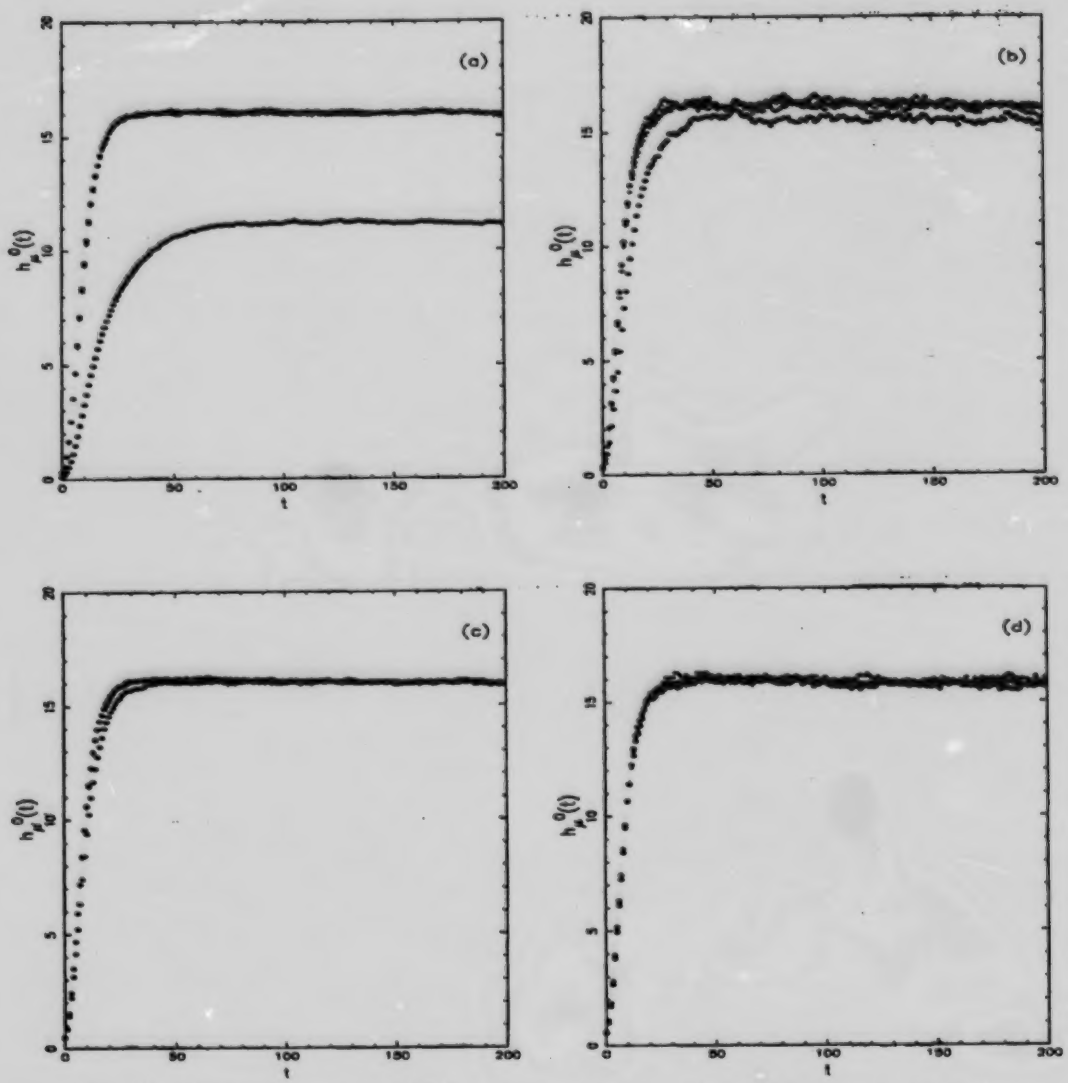

Figure 3.2: The distance $h_\mu^0(t)$ vs. t for $t_l = 1000$ in the modified model with (a) noise 0.1, (b) noise 0.3, (c) noise 0.5 and (d) noise 0.7. The cross-circles, crosses, stars and circles represent the distance between S and ξ_1, ξ_2, ξ_3 and ξ_4, respectively. The later three distances are very close to one another and are evidently larger that $h_1^0(t)$ for noise less than 0.7.

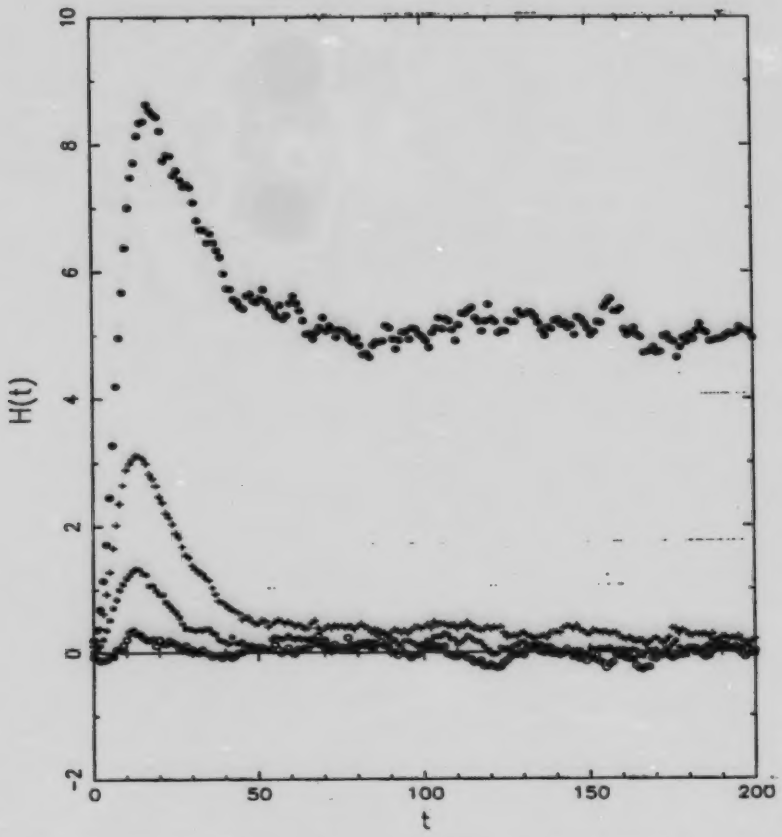

Figure 3.3: The relative hamming-distance $H(t)$ vs. t with $t_l = 1000$. The cross-circles, crosses, stars and circles represent noise 0.1, 0.3, 0.5 and 0.7, respectively.

3.2 The synchronization of chaos in pattern recognition

If we use the absolute hamming-distance to recognize patterns, we have to answer the question proposed in sect. 2.2: how do we explain the fact that on the one hand, the distance increases with the increment of time, and on the other hand, the minimal distance is utilized for recognition? We try to answer it with the approach of synchronization.

As discussed above, we have in the space of states one trajectory yielded by the twisted state and more trajectories yield by the patterns. We refer the former and later trajectories as the "slave" and "master" trajectories with usual terminology in chaos control. The two kinds of trajectories evolve independently, if there is no coupling between them. But if a synchronization between them is introduced, they can overlap after some time. There are many approaches [10, 13] to synchronize the trajectories. Here we give a simple linear feedback synchronization. The point of the slave and the master trajectory at time t are denoted by s_i^t and $f_{\mu i}^t$, respectively as in Eq. (2.5). Before going to the next dynamic evolution, s_i^t is modified to be

$$q_i^t = s_i^t + k(f_{\mu i}^t - s_i^t), \tag{3.2}$$

where $0 < k < 1$ is a constant referred as feedback coefficient. q_i^t is used for next dynamics. After sufficient long time, q_i^t will be identified (be synchronized) with $f_{\mu i}(t)$. This process depends of course on k. For $k = 0$, there is no synchronization; $k = 1$, the slave trajectory is modified immediately to be the master trajectory. Different k will require different time to synchronize. Before the overlap of the two trajectories, the evolution of the slave trajectory is determined by the dynamics (from the system) *and* the feedback procedure. At every time step, the slave trajectory comes a bit closer to the master trajectory, depending on the strength of the feedback term. Eventually, the slave trajectory merges to the master trajectory. After the synchronization, the master and slave trajectories evolve together and the feedback term goes to zero. In the network there are more than one master trajectory which we want to recognize. We require that the synchronization of the slave trajectory with different master trajectories should give a recognizable character or a criterion that tells us which pattern (represented by a

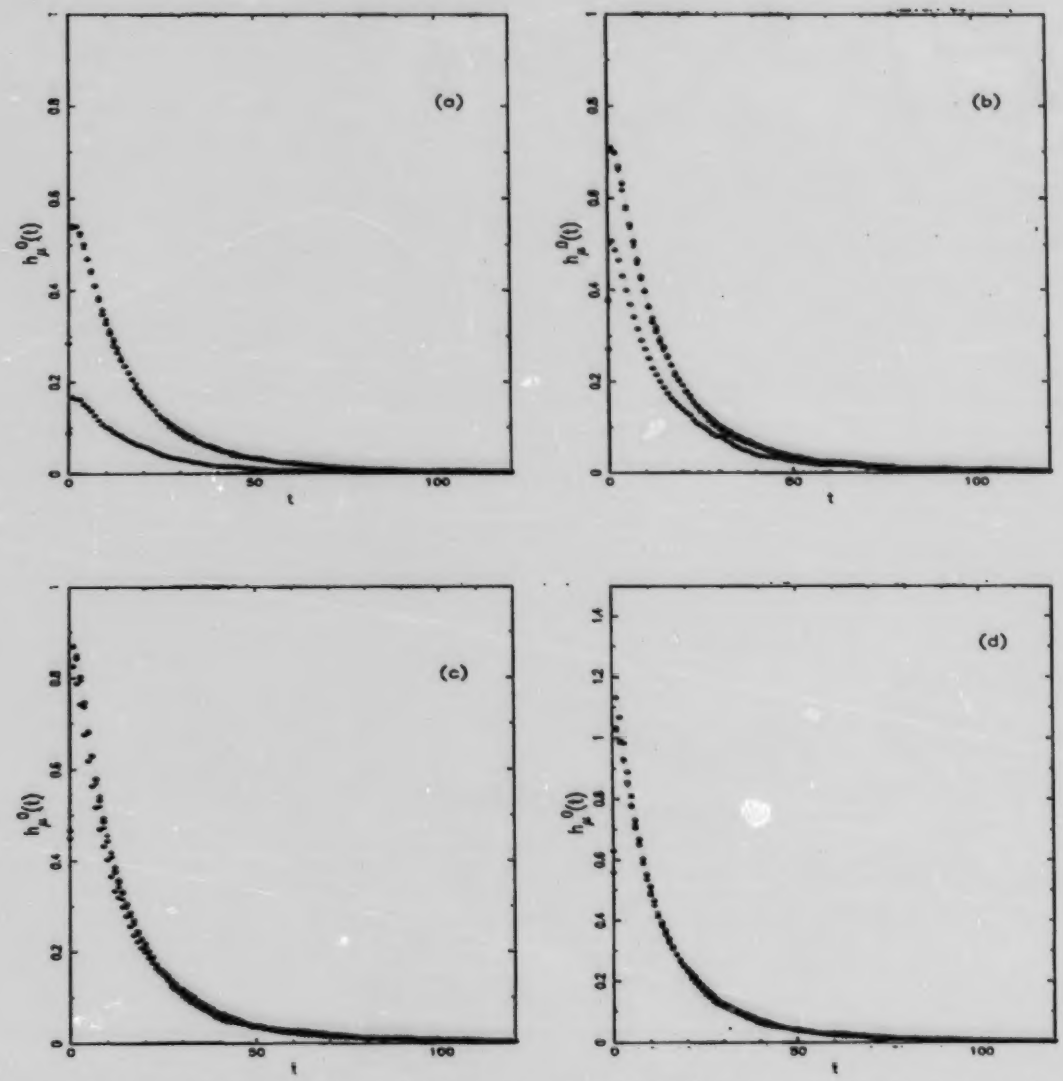

Figure 3.4: The distance $h_\mu^0(t)$ vs. t for $t_l = 1000$ with (a) noise 0.1, (b) noise 0.3, (c) noise 0.5 and (d) noise 0.7. The feedback is introduced with coefficient $k = 0.28$. The cross-circles, crosses, stars and circles represent the distance between S and ξ_1, ξ_2, ξ_3 and ξ_4, respectively. The later three distances are packed together, while $h_1^0(t)$ is distinct from them whenever the system can recognize patterns.

master) the state (represented by the slave) should be recognized to be. We chose the time of synchronization as the criterion. The synchronization with different masters needs different time. We argue that the shortest time gives the master (pattern) with which the slave (state) identifies itself. Further work needs to be done.

Fig. 3.4 gives the synchronized distance for $t_l = 1000$ with noise $0.1, 0.3, 0.5$ and 0.7. All distances are synchronized to be zero after some time. But the minimal distance shown in Fig. 3.2 goes to zero faster than other distances. Therefore the time of synchronization gives the information included in the state. When noise > 0.5, the time of recognition is not distinguishable. This is in agreement with our earlier observation without feedback (Fig. 3.2). The synchronization guarantees a small distance after some dynamic steps and at the same time, it gives the time which can be used to select the needed pattern from more masters. The former property satisfies the requirement that the distance between the pattern and the state should be small, as in the recognition without chaos. The later property tells us how to recognize the patterns with synchronization in the chaotic system.

Chapter 4

Conclusions

The model used by Barton is interesting in that it can be used to recognize patterns, when the dynamics is chaotic or nonchaotic. The recognition has the character of short-term memory, in both the learning process and the recognizing process. The minimal absolute-hamming distance can be identified as the recognition. The synchronization of the chaos control guarantees a small distance between the pattern and the state. The time of synchronization can be utilized to determine the pattern which the state should be recognized to be.

Following studies would be interesting and necessary:

1. The chaos in Barton's model need to be studied further. More exact relation in the short-term memory, such as learning time and recognizing time, should be determined. Specially, the relation with the size of the network and the storage capacity should be clarified. The storage capacity is an important parameter to describe a model. It tell us how many patterns can be stored in a model, before the network fails to work.

2. The short-term memory in other chaotic models. Chaos is existing in many biological models. It is worth to study if the models possess short term memory to process information, such as storing, recognizing and analyzing information.

3. Applying the criterion of relative dynamic hamming-distance to other

models. Because in the chaotic case the system normally does not evolve to fixed points, the information is assumed to be included in trajectories in the space of states, a dynamic quantity is necessary to measure the distance between a "master" trajectory and a "slave" trajectory. The recognition is identified with the existence of a minimal distance. This approach provides in principle a general description. Can it be used in other models, too?

4. The generalization of synchronization to pattern recognition in neural network. Synchronization is a approach discussed intensively in chaos studies. There are many methods to introduce synchronization [10, 11, 12, 13]. The reasonable choice should extract the information hidden in the chaotic dynamics and at the same time, it should not be strong to obscure the pure dynamics of the system. If it can be applied in the pattern recognition, it provides a new approach in neural network. The significance is important.

Bibliography

[1] Barton S A, "The effect of sensory input on trajectories generated by recurrent neural networks", DRES SR-589 (1993)

[2] Barton S A, "The two dimensional movement controlled by a chaotic neural network", Automatica 31, 1149 (1995)

[3] Private communication with Barton S A.

[4] Hopfield J J, Proc. Natl. Acad. Sci. USA Vol. 79, 2554 (1982)

[5] Amit D J, Gutfreund H and Sompolinsky H, Phys. Rev. A32, 1007 (1985)

[6] Amit D J, Gutfreund H and Sompolinsky H, Phys. Rev. Lett. 55, 1530 (1985); Ann. Phys.173, 30 (1987);

[7] Gardner E, J. Phys. A: Math.Gen. 22, 1969 (1989)

[8] Kepler T B and Abbott L F, J. Physique 49, 1657 (1988)

[9] Tan Z and Schülke L, Inter. J. of Mod. Phys. B26, 3549 (1996)

[10] Ali M K and Fang J-Q, Phys. Rev. E55, 5285 (1997)

[11] Ott E, Grebogi C and Yorke J A, Phys. Rev. Lett. 64, 1196 (1990)

[12] Shinbrot T, Ott E, Grebogi C and Yorke J A, Phys. Rev. Lett. 65, 3215 (1990)

[13] Pyragas K, Phys. Lett. A170, 421 (1992)

Report 2

Pattern Recognition in a Neural Network with Chaos

(October - December, 1997)

Contents

1 introduction **3**

2 Chaos in the Gardner model **5**
2.1 The Lyapunov Exponent 6
2.2 The Auto-correlation Function 7

3 Pattern recognition with chaos control **10**
3.1 The Dynamic Distances and the Time of Synchronization . . . 11
3.2 Chaos Control in the Pattern Recognition 14

4 Conclusions **19**

Chapter 1

introduction

Chaos in neural networks has attracted much interests in recent years (see for example [1, 2, 3, 4, 5].). There are beliefs and evidences that chaos might play an active role in the mechanisms of learning and memory retrieval. However, a definitive study on the role of chaos in neural networks is still missing and our goal is towards such a study. As it is well known chaotic systems have three important characteristics: (1) a chaotic system is topologically transitive meaning that every point in phase space can be reached from every other point through dynamical evolution (there is no disjoint invariant open sets), (2) periodic orbits are dense in chaotic systems, and (3) chaotic systems are sensitive to initial conditions (rapid memory loss). Our aim is to see how a chaotic network with such characteristics can be used for pattern recognition.

In our previous studies [6] we have shown that chaos control can be used in pattern recognition in a model used by Barton [7, 8, 9]. That model possesses chaotic or periodic dynamics depending on the parameters involved in it. In that model fixed points are not necessary to perform recognition of patterns. The history of the evolution provides information which can be extracted by investigating a dynamic hamming distance between the master trajectory and the one to be recognized. The feedback is applied to guarantee a small enough hamming distance and a difference among different hamming distances.

In this report, we focus on the Gardner model [10, 11, 12, 13] which is one of the most well-known models in neural networks. Some of the concepts are based on our previous work [6]. Through these studies, we hope to establish

a general method for recognizing patterns in systems with chaotic dynamics. In Chapter 2, we introduce chaos in this model. The Lyapunov exponent and the auto-correlation function are used to show the existence of chaos in the model. In Chapter 3 we discuss pattern recognition, combined with the chaos control. We hope to give an example applying chaos control in the recognition of neural network. Rather than proving whether chaos plays a significant role in neural networks or not, we propose here an approach to recognition of patterns in the Gardner model in the presence of chaos. Conclusions and further research directions are presented in Chapter 4.

Chapter 2

Chaos in the Gardner model

In the Gardner model [10], N neurons are connected to one another through the synapses J_{ij}, $i, j = 1, \ldots, N$. The state, $S_i(t)$, of the i-th neuron at time t is the spin variable $S_i(t) = \pm 1$. The evolution of the state is determined by *all* other neurons through synapses with the dynamics

$$S_i(t+1) = sgn\left(\frac{1}{\sqrt{N}} \sum_{j \neq i}^{N} J_{ij} S_j(t)\right). \tag{2.1}$$

A scale on J_{ij} is imposed

$$\sum_{j \neq i} J_{ij}^2 = N. \tag{2.2}$$

In order to store a number of p patterns, $\xi_i^\mu = \pm 1, \mu = 1, \ldots, p$, the synapses are trained so that the condition

$$\frac{1}{\sqrt{N}} \xi_i^\mu \sum_{j \neq i} J_{ij} \xi_j^\mu > K \geq 0 \tag{2.3}$$

is satisfied. This model can work as an associate memory: it is capable of remembering maximum $2N$ patterns and retrieving these patterns, based on partial information about them [11, 12, 13]. The patterns coded in synapses are the fixed points under the dynamics of the system. The larger is the parameter K, the larger are the basins of fixed points.

Details of the Gardner model are available in the literature. The statistical treatments in the Gardner model can be generalized to obtain many

other spin-glass models of neural networks. Thus the Gardner model is, in fact, the basic and well-known model in this field. In this report, we are interested in chaos in the Gardner model and in recognizing patterns when chaos exists in this model.

2.1 The Lyapunov Exponent

Chaos can be introduced into the Gardner model if a neuron is connected to fewer than $N-1$ of the total N neurons. The number of missing synapses, chosen randomly, is represented as a fraction c of the total $N-1$ synapses for any neuron. One may say that a fraction of the synapses are 'cut off' randomly. A network with partially connected neurons has a similarity with the human brain in which the neurons are not fully connected [14]. The loss of synaptic connections in the human brain may also occur because of accidents or old age [2]. In the Gardner model, the cutting off of synapses (CUTS) has been discussed earlier as a diluted approach [15], where the effect of the dilution on the retrieval of patterns hidden in the fixed points were considered. However, the existing work concentrates exclusively on the concept of pattern recognition being persistent under dynamics without mentioning its relation to chaos. In ref. [5] chaos is introduced in a similar way to a recurrent neural network with the Hebbian-like learning rule and a "cycle of memory" is discussed.

The usual way to identify chaos in a network is to calculate the Lyapunov exponents or auto-correlation functions [16, 17]. In this section we discuss the largest Lyapunov exponent and leave the discussion of the auto-correlation function for the next section. The numerical method to get the largest Lyapunov exponent is [17]

$$\sigma = \lim_{n\to\infty} \frac{1}{n\tau} \sum_{i}^{n} ln\left(\frac{d(i\tau)}{d(0)}\right), \tag{2.4}$$

where $d(0)$ and $d(i\tau)$ are the distances between two states at time $t=0$ and $t=i\tau$, respectively. σ is independent of the interval τ which should not be too large. The criterion tells us that if σ is positive, the system is chaotic.

Table (2.1) contains σ for various values of K and c, where $0 \leq c \leq 1$ is the fraction of synapses removed randomly from a neuron. Table (2.1)

Table 2.1: The Lyapunov exponent σ

K	c	σ
2	0.	-0.3
2	0.2	-4.5E-4
2	0.5	0.1
2	0.8	0.29
1.5	0.5	0.28
0.68	0.5	0.29

shows that the largest Lyapunov exponent becomes positive as the fraction of CUTS increases showing the presence of chaos. The table also shows that σ depends on K, the basin of fixed points defined in original (fully connected) model. When K is large, the convergence of a state to a fixed point is strong (strong convergent case). Small dilution of the synapses will not influence the convergence significantly. The model with slight CUTS retains the main property of the fully connected model. However, when the fraction, c, is large, the network exhibits chaotic features, no matter how strong is the convergence. We now look at the auto-correlation function as a measure of chaos.

2.2 The Auto-correlation Function

The auto-correlation function [16, 17] $C(\tau)$ is defined by

$$C(\tau) = \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T} S_i(t) S_i(t+\tau) \tag{2.5}$$

where $S_i(t)$ is the state of neuron i at time t. $C(\tau)$ is independent of the choice of i. The criterion of the auto-correlation function states that if $C(\tau)$ is a decreasing function of τ, the system is chaotic. The speed at which $C(\tau)$ decreases is related to the rate which the systems turns chaotic. Table (2.2) gives the auto-correlation function $C(\tau)$ as a function of τ for various values of c and $K = 2$. The dependence of $C(\tau)$ on τ is also presented in Fig. 2.1. When $c = 0$, $C(\tau)$ remains as a constant while for $c > 0$, $C(\tau)$ decreases

Table 2.2: The auto-correlation function at $K = 2$

τ	$C(\tau)_{c=0}$	$C(\tau)_{c=0.5}$	$C(\tau)_{c=0.6}$	$C(\tau)_{c=0.8}$
0	1.	1.	1.	1.
1	1.	0.72	0.34	0.09
20	1.	0.64	0.065	0.0003
40	1.	0.59	0.013	0.
100	1.	0.48	0.	0.
200	1.	0.35	0.	0.
400	1.	0.19	0.	0.
1000	1.	0.04	0.	0.
2000	1.	0.	0.	0.

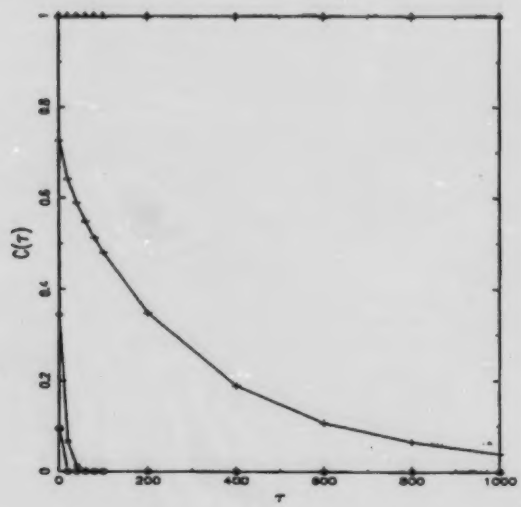

Figure 2.1: The auto-correlation function of the Gardner model at $K = 2$ for $c = 0$ (triangles), 0.5 (crosses), 0.6 (stars) and 0.8 (circles). Points are linked for a better perception.

from a finite value to zero. For larger values of c, $C(\tau)$ goes to zero faster. It should be pointed out that $C(\tau)$ is also dependent on K. For smaller K, even a small c will produce chaos.

From the largest Lyapunov exponent and the auto-correlation function, we conclude that chaos is introduced into the Gardner model by cutting off synapses randomly and the chaos depends on c and K. The next question is: can the partially connected chaotic Gardner network recognize patterns that were stored in the fully connected network? We look for an answer to this question in the next chapter.

Chapter 3

Pattern recognition with chaos control

When chaos is introduced, dynamics will not drive an initial state to a fixed point (pattern) as in the nonchaotic model without CUTS. In the chaotic case, condition (2.3) is violated and patterns ξ^μ are no more the fixed points. In fact, a chaotic system has infinite number of unstable period orbits (UPO) [18, 19]. But we believe that our network still has some memory (at least when K is large) before it manifests complete chaotic behavior. The synapses are trained before the CUTS and therefore, we believe that some information about the patterns may still be present in the J_{ij}. Before a fully chaotic behavior sets in, it may be possible to use the network for pattern identification. Here, we are referring to the short-term memory. The problem now is to extract this information, if there is any. To this end, we turn to chaos control. It is well known that chaotic trajectories can be synchronized (stabilized) to desired trajectories. Synchronization of trajectories may provide [20] an effective method which is completely different from the usual concepts in pattern recognition. The approach to synchronization of chaotic trajectories is not unique [18, 19]. Here we utilize the feedback method proposed in [18] and select the trajectory stimulated initially by a pattern (e.g. ξ^1) as desired orbit and synchronize other chaotic trajectories to this orbit. We try to get useful information during the synchronization.

3.1 The Dynamic Distances and the Time of Synchronization

We denote the desired trajectory as $\zeta^1(t)$ with $\zeta^1(0) = \xi^1$ and a trajectory to be synchronized as $s(t)$ which is near ξ^1 at $t = 0$. The relation between $\zeta^1(t)$ and $s(t)$ is measured by a dynamic distance

$$d^1(t) = \left\langle \frac{1}{N} \sum_{i=1}^{N} (s_i(t) - \zeta_i^1(t))^2 \right\rangle_\xi , \tag{3.1}$$

where, $\langle \cdots \rangle_\xi$ means the average over random patterns ξ in the Gardner model [10, 13]. $d^1(0)$ gives the initial distance. Although $\zeta^1(t)$ is not ξ^1, it is generated by ξ^1 and in a deterministic system it is unique and repeatable (even though the trajectories depend on the machine and numerical accuracy) if started again from ξ^1. Therefore, we "transfer" ξ^1 to $\zeta^1(t)$. $s(0)$ is recognized to be ξ^1 if $d^1(t)$ goes to zero (or a small value).

In this chaotic model with larger c, the dynamics will drive $s(t)$ away from $\zeta^1(t)$. $d^1(t)(t > 0)$ is larger than $d^1(0)$. But if at every dynamic step we "help" $s(t)$ to go a little bit closer to $\zeta^1(t)$, we may get useful results. This is just the idea of feedback of chaos control. We now turn to chaos control and try to perform recognition in the chaotic system.

The feedback is imposed by modifying the trajectory $s(t)$ at time t with

$$s_i(t) \to s_i^f(t) = s_i(t) + k(\zeta_i^1(t) - s_i(t)), \quad i = 1, \ldots, N \tag{3.2}$$

where $0 < k < 1$ is a constant called hereafter as the feedback strength. For $k = 0$, there is no feedback; $k = 1$, the trajectory is modified immediately to be the desired one. The state is $s^f(t)$ after the feedback process is utilized for the next dynamics. Because the feedback is applied directly and at every time step on the output of the system, $s^f(t)$ will be driven to $\zeta^1(t)$ after sufficiently long time depending on k. After the synchronization, the feedback term goes to zero and the distance in Eq. (3.1) vanishes (or becomes very small). However, we cannot believe that $d^1(t) = 0$ in this case means that $s(t)$ is recognized, because $s(t)$ can be synchronized in the same way with all other trajectories $\zeta^2, \ldots, \zeta^p$ coming initially from other patterns, $\xi^2, \ldots, \xi^p$,

respectively. Therefore, we need another criterion to distinguish between the synchronized orbits.

The criterion that we have used in this work is the time of synchronization. We believe that the shortest time of synchronization tells us which pattern the state $s(0)$ should be identified with. More exactly, we denote $d^\mu(t)$ as the distance by replacing index 1 in Eq. (3.1) with μ. The shortest time

$$t^r = min\{t^1, \ldots, t^p\}, \tag{3.3}$$

where t^μ satisfy

$$d^\mu(t^\mu) < d_0, \quad \mu = 1, \ldots, p, \tag{3.4}$$

determines that pattern r is the target pattern and d_0 is a small number. If the shortest time t^r does not exist ($d^\mu(t^\mu)$ are same for all patterns before d_0 is reached), then $s(0)$ can not be recognized. In practice we do not have to compare all distances, because after the average in Eq. (3.1) all patterns are equivalent. Below we set $s(0)$ to be related to ξ^1 with the initial distance $d^1(0)$ and compare the distance between $s(t)$ and $\zeta^1(t)$ and that between $s(t)$ and an arbitrary trajectory, see $\zeta^2(t)$.

We first check the method with the fully connected Gardner model ($c = 0$) without chaos. Fig. 3.1 (a) and Fig. 3.1 (b) gives the distances $d^1(t)$ and $d^2(t)$ versus t with the initial distance $d^1(0) = 1$ for $K = 2$ and for $K = 0.68$, respectively. In both cases we set $d_0 = 0.02$, the feedback strength $k = 0.04$ in (a) and $k = 0.3$ in (b). In (a) $d^1(t)$ is clearly smaller than $d^2(t)$ and this leads to the result that t^1 is smaller than t^2 (actually t^2 is out of the range shown in the figure) when condition (3.4) is satisfied. In (b) $d^1(t)$ and $d^2(t)$ have the same value after certain steps and this will not give any difference between t^1 and t^2. Therefore, we say that pattern recognition is realized in case (a) but not in case (b). This is in agreement with the well-known results obtained by means of fixed points [10, 12, 13] as shown in Fig. 3.1 (c) and (d), where the feedback is not imposed and pattern recognition is processed based on the criterion of comparing the value of $d^1(t)$ at $t > 0$ and the value at $t = 0$. Because the Gardner model is actually a *static* model, pattern will be recognized if $d^1(t > 0) < d^1(0)$ (as shown in (c)). Otherwise if $d^1(t > 0) > d^1(0)$ pattern cannot be recognized (as shown in (d)).

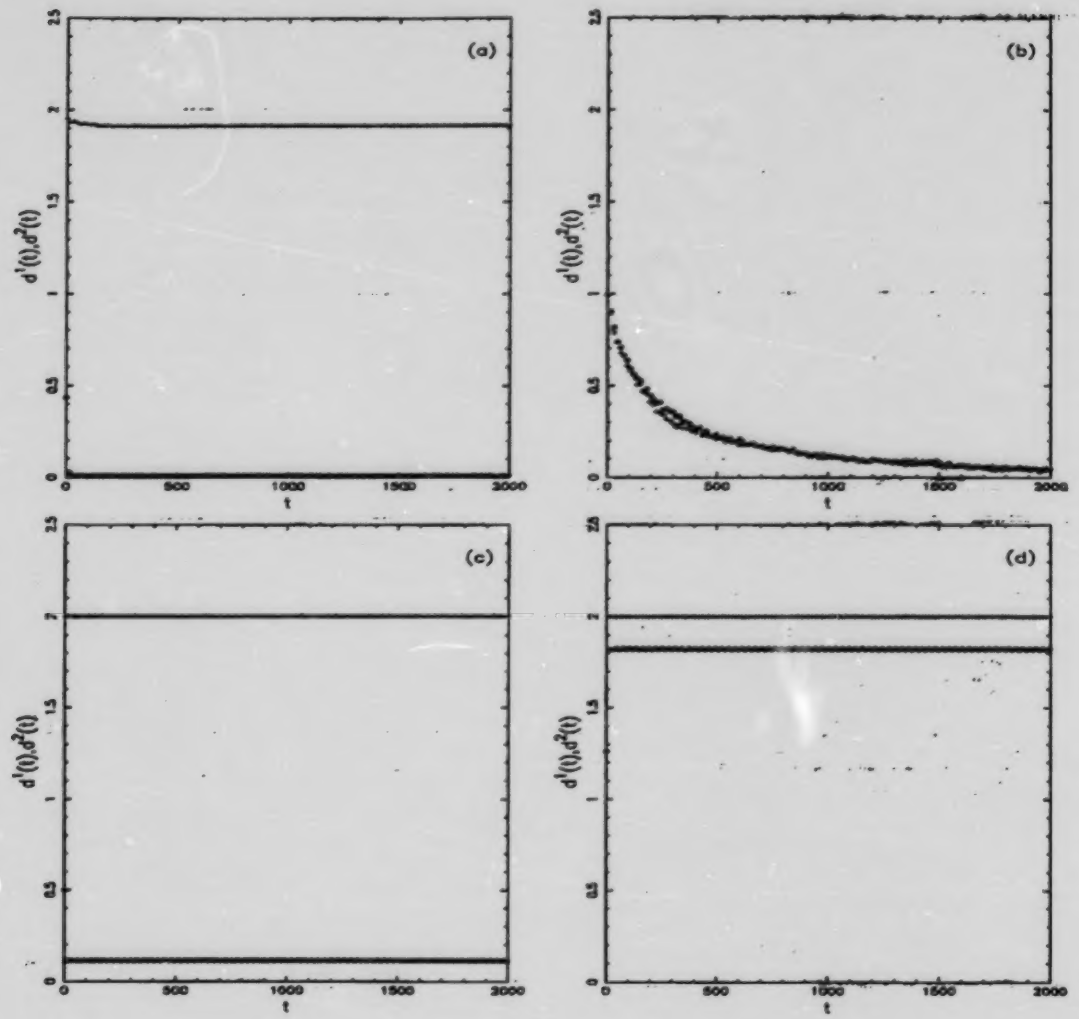

Figure 3.1: The dynamic distances $d^1(t)$ (circles) and $d^2(t)$ (crosses) vs. time t in the previous Gardner model with $d^1(0) = 1$ and $d_0 = 0.02$. (a)$K = 2$, feedback strength $k = 0.08$. Pattern is recognized. (b)$K = 0.68$, feedback strength $k = 0.02$. Pattern is failed to be recognized. (c) and (d) correspond to (a) and (b), respectively, but no feedback is imposed.

It may be pointed out that the qualitative property is not sensitive to the feedback strength k. k determines the time required for satisfying the condition (3.4) and does not affect the order in magnitudes among different t^μ. We would like to show this in the following sections.

3.2 Chaos Control in the Pattern Recognition

We return to the chaotic model. It is natural that when c is small, the structure of the fully connected model will not be affected dramatically. Specially when the fixed points have larger basin (e.g. $K = 2$), the strong convergent tendency persists. Our results verify this behavior. When c is large, things change noticeably. Fig. 3.2 gives the dynamic distance $d^1(t)$ (circles) and $d^2(t)$ (crosses) versus t for $K = 2$, $d^1(0) = 1$, $d_0 = 0.02$ and various fractions of CUTS and feedback strengths. In Fig. 3.2(a), c has a small value 0.2, the cutoff will not introduce fully developed chaos into the convergent dynamics. The state near a pattern will converge to the pattern. Therefore, even without feedback $d^1(t > 0)$ is smaller than the initial distance $d^1(0)$ and this is just the case of the Fig. 3.1(c) discussed above. In Fig. 3.2(b) the fraction of CUTS increases to $c = 0.4$ and the chaos is thus stronger than that in Fig. 3.2(a). When the feedback strength $k = 0.04$ is used, $d^1(t)$ goes to zero much faster than $d^2(t)$ and this implies that t^1 is smaller that t^2 when condition (3.4) is fulfilled. In Fig. 3.2(c) with $c = 0.5$, it is not easy to distinguish between t^1 and t^2 from the curves. When $c = 0.55$ in Fig. 3.2(d), we cannot find any difference between $d^1(t)$ and $d^2(t)$ after certain steps. This will not lead to different t^1 and t^2. Therefore, the pattern is recognized in Figs. 3.2(a) and (b) but not in Figs. 3.2(c) and (d).

In order to show the influence of the feedback strength k on the order of synchronization times, we have used a number of values of k in Fig. 3.2. It can be seen that there is no change in the order among t^μ i.e. if t^r is the smallest for a given feedback strength, it will still be the smallest for other choices of the strength.

Other factors to influence the synchronization are the basin parameter K and the initial distance $d^1(0)$. Fig. 3.3(a) shows the effect of K. Compared

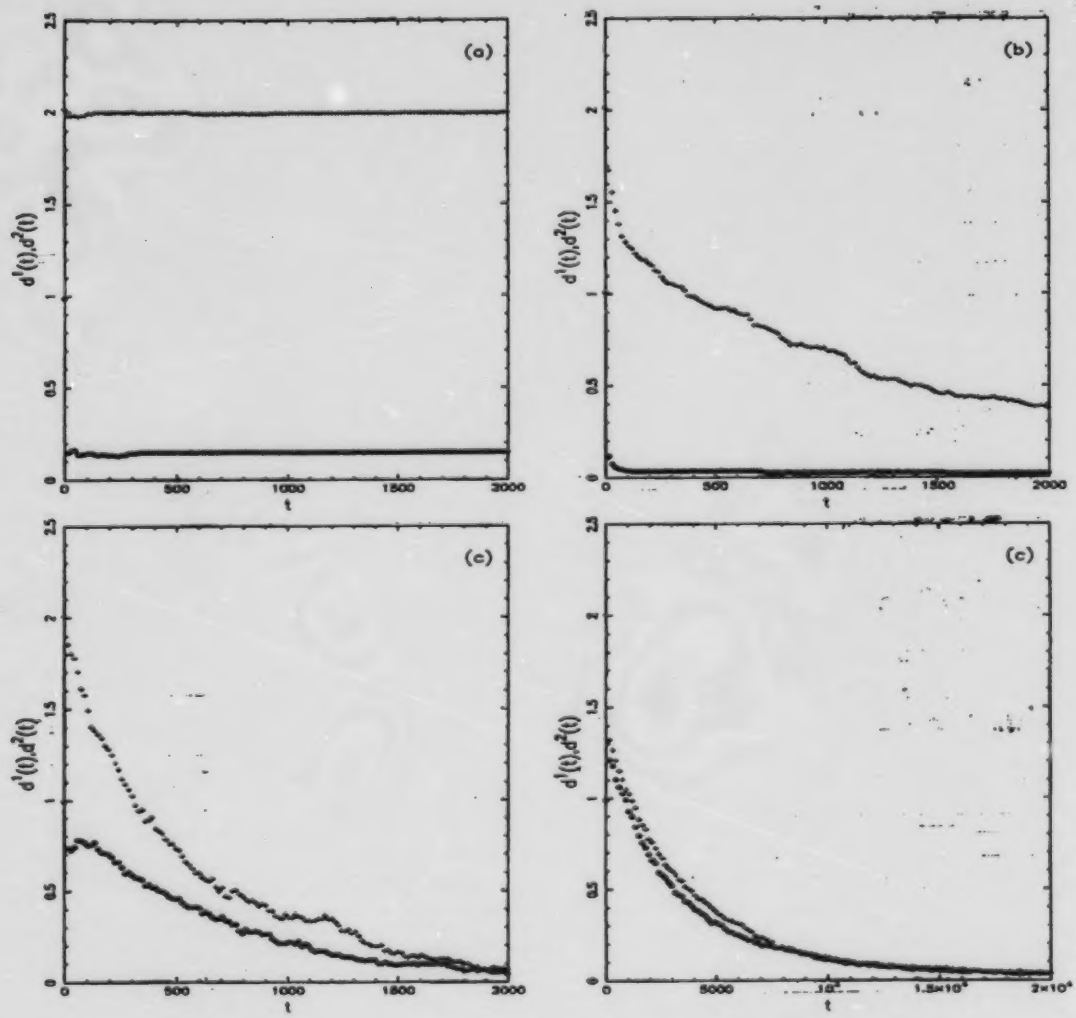

Figure 3.2: The dynamic distances $d^1(t)$ (circles) and $d^2(t)$ (crosses) vs. time t for $d^1(0) = 1$, $d_0 = 0.02$, $K = 2$ and (a)$c = 0.2$, $k = 0$; (b)$c = 0.4$, $k = 0.04$; (c)$c = 0.5$, $k = 0.02$; (d)$c = 0.55$, $k = 0.06$. In case (a) and (b) patterns can be recognized, while in case (c) and (d) it is failed to recognize patterns.

with $K = 2$ in Fig. 3.2(b), Fig. 3.3(a) gives the result for $K = 1.5$ with the same value $c = 0.4$ and initial distance. Clearly, the two distances after certain steps cannot be distinguished. In Fig. 3.3(b) we show the effect of $d^1(0)$. Comparing Fig. 3.3(b) and Fig. 3.2(c), we have just changed $d^1(0)$. The difference between t^1 and t^2 in Fig. 3.3(b) is clear, while in Fig. 3.2(c) it is much more difficult to see such difference. Therefore, the ability of recog-

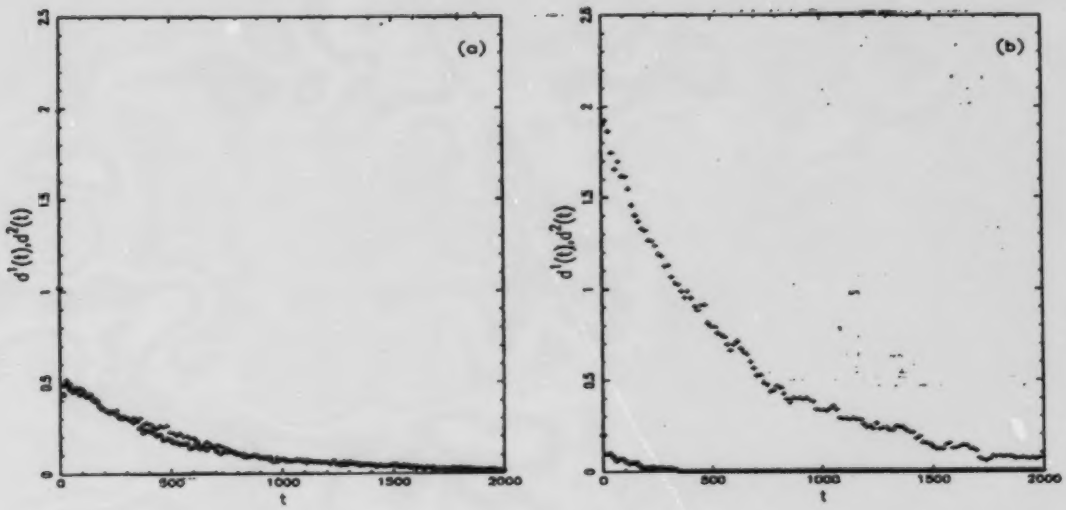

Figure 3.3: The dynamic distances $d^1(t)$ (circles) and $d^2(t)$ (crosses) vs. time t for (a) $K = 1.5$, $c = 0.4$, $k = 0.35$ and $d^1(0) = 1.$; (b) $K = 2$, $c = 0.5$, $k = 0.02$ and $d^1(0) = 0.2$

nition dependents on the parameters K (which determines the basins of the fully connected nonchaotic model), c (which defines the fraction of CUTS), and $d^1(0)$ (which is the initial distance). Table 3.1 gives for $K = 2$ and certain $d^1(0)$ the critical values of c below which patterns can be recognized. We see from the table that for larger K and smaller initial distance $d^1(0)$, larger numbers of synapses can be removed and patterns can still be recognized. When the convergent tendency is not strong, the dynamics is affected easily by the chaos. Therefore only for very small initial distances, we can recognized patterns. If the fraction of CUTS is reduced to 0.2 with $K = 1.5$,

Table 3.1: Critic c below which patterns can be recognized

$d^1(0)_{K=2}$	$d^1(0)_{K=1.5}$	c
0.2	0.14	0.52
1.0	0.15	0.5
1.6	0.15	0.45

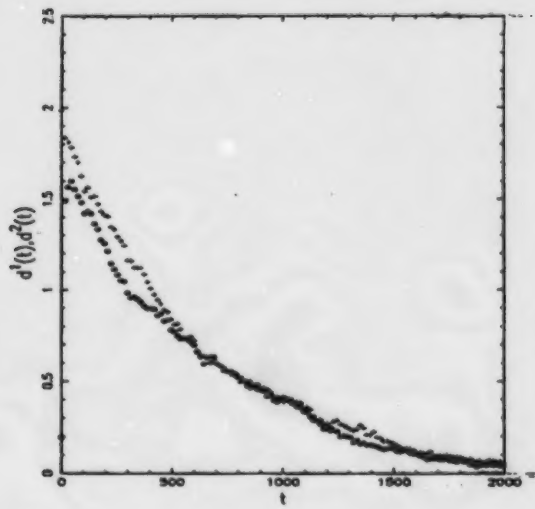

Figure 3.4: The dynamic distances $d^1(t)$ (circles) and $d^2(t)$ (crosses) started from two random states for $K = 2$, $d^1(0) = 0.2$, $c = 0.5$ and $k = 0.02$. The distances are indistinguishable after certain steps.

$d^1(0)$ can increase to 1.4 (not shown here).

We emphasize that this result is not only related to the initial condition, but also to the patterns which have been embedded in J_{ij}. This can be verified with starting ζ^1 and ζ^2 by two random states instead of ξ^1 and ξ^2. Fig 3.4 gives the distance $d^1(t)$ between $\zeta^1(t)$ and $s(t)$ and the distance $d^2(t)$ between $\zeta^2(t)$ and $s(t)$. Even for very small initial distance ($d^1(t) = 0.2$) and strong convergent tendency ($K = 2$) we cannot distinguish $d^1(t)$ and $d^2(t)$ as we did in Fig. 3.3(b). This is not very surprizing, because the initial random states are not stored in the J_{ij}.

Chapter 4

Conclusions

We have tried to recognize patterns in the chaotic Gardner model by means of chaos control. This approach does not need stable fixed points which do not exit in the chaotic model. The chaotic model stores information in the synapses for a short term. Using feedback to synchronize two trajectories representing the patterns and states to be recognized, we introduce the time of synchronization as our criterion. The shortest time gives us the information about patterns. The synchronization depends on c (the fraction of CUTS), K (the basin parameter) and $d^1(0)$ (the initial condition).

The manner in which the synapses are cut off may be important. If the synapses are reduced slowly, the results may be different. In this report we have discussed the "abrupt" cutting in which the synapses are reduced to a certain fraction at the very beginning. The structure of the network is changed suddenly and a lot of information of the fully connected network has been lost . If the rate, at which the CUTS is carried out, is slow, we may observe a gradual change of the network with more memory retained. This may help us, for example, to discuss the metastable states which capture the flow of dynamics. As in Fig. 3.1(c) the distance keeps to be constant after a few steps. A perturbation by CUTS could help the flow escape the metastable states, while the main structure is kept. Conversely, we can also discuss the restoring of synapses back to the fully connected form. Work on these aspects is interesting and is in progress.

We have observed another interesting phenomenon which is related to the

long time dynamics. For certain values of c, both $d^1(t)$ and $d^2(t)$ will go to zero after long enough time *without feedback*. For example, when $K = 2$, $d^1(0) = 1$ and $c = 0.5$, $d^1(t)$ goes to zero after about 5×10^5 time steps and after some othe time steps $d^2(t)$ goes to zero, too. For $c = 0.55$, the same behavior is observed except that a longer time is required. It seems that there is an attractor which has a very large basin. All states will be attracted to this basin eventually. The information about the different states exists only before they are driven to this basin. The meaning of the behavior needs to be studied further.

So far, we have not used the infinitely many unstable periodic orbits for pattern recognition. We plan to extend our work in this important direction.

Bibliography

[1] Skarda C A and Freeman W J, Behavior and Brain Science 10, 161 (1987).

[2] Glanz J, Science, Vol. 277, 1758 (1997).

[3] Tsuda I, Neural Networks, Vol. 5, 313 (1992).

[4] Matsumoto K and Tsuda I, J. Phys. A18, 3561 (1985).

[5] Nara S and Davis P, Prog. Theor. Phys., Vol. 88, 845 (1992).

[6] Tan Z and Ali M K, Progress Report to DND (Sept. 1997)

[7] Barton S A, "The effect of sensory input on trajectories generated by recurrent neural networks", DRES SR-589 (1993)

[8] Barton S A, "The two dimensional movement controlled by a chaotic neural network", Automatica 31, 1149 (1995)

[9] Private communications with Barton S A.

[10] Gardner E, J. Phys. A21, 257 (1988).

[11] Gardner E, J. Phys. A22, 1969 (1989).

[12] Kepler T B and Abbott L F, J. Physique 49, 1657 (1988).

[13] Tan Z and Schülke L, Inter. J. Mod. Phys. B26, 3549 (1996).

[14] Clark J W, Phys. Rep. 158, 91 (1988).

[15] Bouten M, Engel A, Komoda A and Serneels R, J. Phys. A23, 4643 (1990).

[16] Milonni P W, Shih M-L and Ackerhalt J R, *Chaos in Laser-Matter Interaction*, World Scientific Publishing Co Pte Ltd. (1987).

[17] Lichtenberg A J and Lieberman M A, *Regular and Stochastic Motion*, Springer-Verlag (1983).

[18] Pyragas K, Phy. Lett. A170, 421 (1992).

[19] Ali M K and Fang J-Q, Phys. Rev. E55, 5285 (1997)

[20] Hoff A A, Z. Naturforsch. 49a, 589 (1994).

Report 3

The Associative Memory in Chaotic Neural Networks

(January - March, 1998)

Contents

1 Introduction 3

2 A Chaotic Recurrent Neural Network with Refractoriness 6
- 2.1 The Refractory Process 6
- 2.2 The Chaotic Recurrent Model 7
- 2.3 The Associative Memory 10

3 Memory Searching in a Network with Synapse Reduction 15
- 3.1 The Model and Chaos 15
- 3.2 Memory Searching 17

4 Dynamic Link of Memory 20
- 4.1 A Model Composed of Two Modules 20
- 4.2 Association and Dynamic Link 22

5 Chaos Control 23
- 5.1 Parameter Control 23
- 5.2 Trajectory Control 28
  - 5.2.1 The Gardner Model 28
  - 5.2.2 Quadratic and Cubic Maps and Continuous Hopfield Model 35

6 Pattern Recognition by Searching the Periodic Orbits 42
- 6.1 The Model Dynamical System 42
- 6.2 Pattern Recognition 44

7 Conclusions 48

Chapter 1

Introduction

Chaos may be unavoidable or may even be necessary in real biological systems [1, 2]. Recently, studies of chaos in neural networks have drawn attention from workers in diverse fields [3]. In this report we focus on the associative memory of chaotic neural networks. Associative memory is an important aspect of a neural information processing system [4, 5, 6, 7, 8], and allows pattern recognition from incomplete presentation of a pattern. We present schemes by which chaotic networks can be used in pattern recognition.

First, we recall the following basic features of chaotic systems: (1) In a chaotic system many chaotic attractors may coexist. The basins of these attractors may be riddled. (2) A chaotic attractor is topologically transitive meaning that every point in the state space can be reached from every other point through dynamical evolution. The frequency at which a randomly initiated trajectory visits the regions of an attractor depends on the dynamical properties of all of the embedded unstable periodic orbits. (3) Periodic orbits are dense in chaotic systems. (4) Chaotic systems are sensitive to initial conditions. This sensitivity implies that chaotic systems lose memory rapidly.

Pattern recognition using chaotic neural networks with such features appears counter intuitive from the point of view of usual pattern recognition methods in neural networks [6, 7, 8, 9, 10, 11] . The usual way of performing pattern recognition in neural networks involves two main steps, (a) the network is taught a certain number of patterns (this determines the synapses) and (b) unknown states which contain partial information of the learned

patterns are recognized as the fixed points reached by running the dynamics. In hard chaos, there are no stable fixed points with attracting basins. Hence, dynamics as such will not drive an arbitrary state to any unstable fixed point as it does in the usual neural network pattern recognition. Thus pattern recognition using chaotic neural networks requires techniques that are either different from or generalizations of those used in nonchaotic systems. If the new techniques are generalizations of concepts that are widely used in nonchaotic cases, one gains the advantage of using chaotic or nonchaotic networks as needed. We present below two general approaches for pattern recognition in neural networks. One of these approaches is based on synchronization of trajectories and the other on methods for finding periodic orbits of chaotic systems.

In order to help motivate the future directions of our work and also to discuss our recent findings in light of the current activities in associative memory in chaotic neural networks, we review some widely used models of such networks. We hope to present a more detailed review of associative memory in chaotic neural networks in the future.

Work on associative memory may be divided into two categories: (1) one that deals mainly with the models themselves and (2) the other that is related to general techniques of pattern recognition. The works of [12] – [21] are representatives of the first category, while those of [22, 23, 24] belong to the second. In this report, first we review the models and then the approaches used in them. In chapter 2, we discuss a recurrent chaotic model proposed by Aihara *et al* [12], based on biological arguments. In this chapter, we discuss the employed method of memory search, a process that is of interest to many people [13, 15, 16]. Then the application of the model in optimization and speech recognition is mentioned. Chapter 3 is about the chaotic model proposed by Nara *et al.* They discuss the "cycle of memory" in the context of the process of memory searching. Chapter 4 reviews the model proposed by Tsuda, which supposes that chaos may be a novelty filter and dynamic link. Chapter 5 deals with the second type of work on associative memory. In this chapter we study the parameter and feedback controls of chaos in various models such as the chaotic Hopfield model, the Gardner model, and the Logistic map. In chapter 6 we present our recent work that introduces a novel approach – a root finder – to discuss associative memory.

Through the root finder, a virtual basin is imposed around a unstable fixed point in a chaotic system, which can be used to perform pattern recognition. The methods utilized in chapter 5 and 6 can in principle be used any chaotic neural network. Finally, the conclusions are presented in chapter 7.

Chapter 2

A Chaotic Recurrent Neural Network with Refractoriness

A model discussed widely is the chaotic recurrent model proposed by Aihara *et al* [12]. In this model, feedback and refractory processes are taken into account. Therefore, we first discuss briefly the background of the refractory process in neurons of real biological systems. The association aspect will be discussed after the introduction.

2.1 The Refractory Process

The refractory process describes the effect of the motion of ions in *past* time on the motion of the ions at the *current* time. There is biological evidence for this phenomena [4]. A real neuron has a membrane that possesses negative voltage. Inside and outside the neuron there are different concentrations of sodium and potassium ions. The flow of ions is determined by their concentration gradients and the electrical potential. Equilibrium of flow of the potassium ions can be reached by these two processes. However, equilibrium of the sodium ions can be broken by a depolarization through adding a positive external stimulus (voltage or current) to the neuron. Such depolarization changes the concentration of ions inside and outside the neuron. When the stimulus is above a threshold, the neuron generates a spike action potential (firing). After each firing, a neuron will go back to the original equilibrium, until the next firing. The frequency of the firing, not the strength of the

firing, is a function of the external stimulus. The concept of refractoriness means that for a brief period it is *not* possible to evoke a second action potential. There is a quiet period (relative refractory period), in which the neuron will not fire. The quiet period is a consequence of the earlier firing (the motion of ions) i.e. the past motion of the ions influences the actual and post movement.

People believe that refractoriness is an important principle for the modelling of artificial neural networks. We will see in the next section one proposal for taking this mechanism into account.

2.2 The Chaotic Recurrent Model

A naive model of a *single* refractory neuron assumes that the refractory process decreases exponentially with time and the dynamics of the neuron can be written

$$s(t+1) = f(A(t) - \alpha \sum_{r=0}^{t} k^r s(t-r) - \theta), \tag{2.1}$$

where $s(t)$ is the state of the neuro at time t, $A(t)$ is the input and the refractory factor, k^r with $0 < k < 1$, decreases exponentially with time. f is the action potential and can be given in different forms.

Eq. 2.1 can be reduced to

$$s(t+1) = f(y(t+1)) \tag{2.2}$$

with

$$y(t+1) = ky(t) - \alpha f(y(t)) + a(t) \tag{2.3}$$

and

$$a(t) = A(t) - kA(t-1) - (1-k)\theta. \tag{2.4}$$

The neuron has chaotic behaviour as can be verified by checking the Lyapunov exponent. Fig. 2.1 gives the exponent as a function of α, using the parameters and potential given in the caption. Over some range of α, the exponent is positive implying the existence of chaos. From the complicated form of the figure one should expect rich dynamic behaviour. Such single

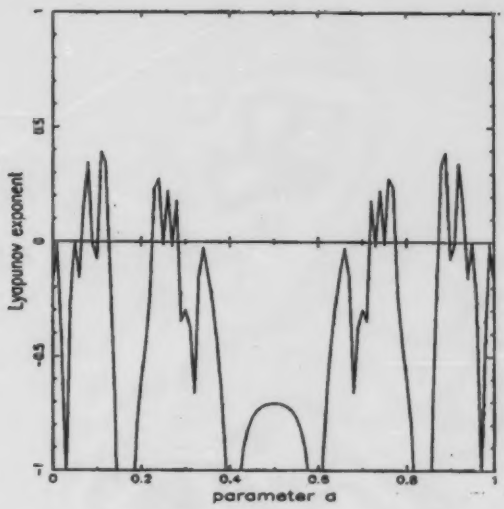

Figure 2.1: The Lyapunov exponent of the dynamics of single neuron with refractory process. $y(0) = 0.5, k = 0.5, \alpha = 1.0$ and $f(y) = (1 + exp(-y/0.04))^{-1}$.

neurons can be used to compose networks according to the following equations [12]

$$s_i(t+1) = f(\eta_i(t+1) + \zeta_i(t+1)), \tag{2.5}$$

$$\phi_i(t+1) = k_e\phi(t) + \sum_{j=1}^{M} v_{ij}A_j(t), \tag{2.6}$$

$$\eta_i(t+1) = k_f\eta_i(t) + \sum_{j=1}^{N} w_{ij}s_j(t), \tag{2.7}$$

$$\zeta_i(t+1) = k_r\zeta_i(t) - \alpha s_i(t) + a_i, \tag{2.8}$$

$$f(z) = [1 + \exp(-\frac{z}{\epsilon})]^{-1}, \tag{2.9}$$

where ϵ is a positive constant determining the steepness of the potential function. The notions in (2.5 – 2.9) must be explained. $s_i(t)$ is the output of neuron i at time t. $\phi_i(t)$ is the external input to the neuron i at time t with v_{ij} the connection weight from the jth external input to the ith neuron.

There are M neurons involved in the communication with the outside. $\eta_i(t)$ is the input from all other neurons connected by the synapses w_{ij} (defined below) with a feedback process of strength k_f. $\zeta_i(t)$ represents the refractory mechanism of neuron i at time t and a feedback of strength k_r and α is also carried out in this process: the neuron will generate a action potential through $f(z)$, when the accumulated feedback terms in $\zeta_i(t)$ exceed the threshold a_i. The threshold a_i, the feedback refractory strength k_r and α determine that neuron i will not stay in one state for long as it makes the neuron change constantly. Even when the initial values of s_i, η_i and ζ_i are null, the nonzero threshold will yield nonstatic output from neuron i. Therefore, the introduction of ζ_i generates chaotic behavior, while the η_i terms produce the usual characteristics of the nonchaotic model. The competition between these two forces results in rich dynamic behavior. This behavior depends of course on the parameters involved in eqs. (2.5-2.9). In order to simplify the discussions, we ignore the contribution from $\phi_i(t)$. This assumes the external input is constant. The contribution will be restored later, when we discuss pattern recognition.

The synapses connecting neurons are defined by the Hebbian rule

$$w_{ij} = \frac{1}{2p} \sum_{\mu=1}^{p} (\xi_i^\mu - \overline{\xi})(\xi_j^\mu - \overline{\xi}), \tag{2.10}$$

where, $\xi_i^\mu, \mu = 1, \ldots, p$, are the patterns stored in this network. They are assigned randomly to be 0 or 1. $\overline{\xi}$ is set to be 0.5.

Fig. 2.2 shows such behaviors for parameters, by giving initial $s_i(t), \eta_i$ and ζ_i zero. For the parameters used in the figure, Fig. 2.2(a) presents chaotic behavior, while Fig. 2.2(b) shows a stable output (that is 1). Therefore, the neuron has chaotic output in Fig. 2.2(a), but not in Fig. 2.2(b). Actually, the property is proven by Ikeguchi *et al* [15], after calculating the Lyapunov exponents in this system. They found that the largest exponents is positive for a network with 16 neurons with $N = 16, p = 4$ and the parameter used in Fig.2.2(a).

The effect of the threshold a_i and feedback parameter α is also related to the dynamic property of the network. One can find the details in the literature. In short, the dynamic behavior depends on the parameters. A typical

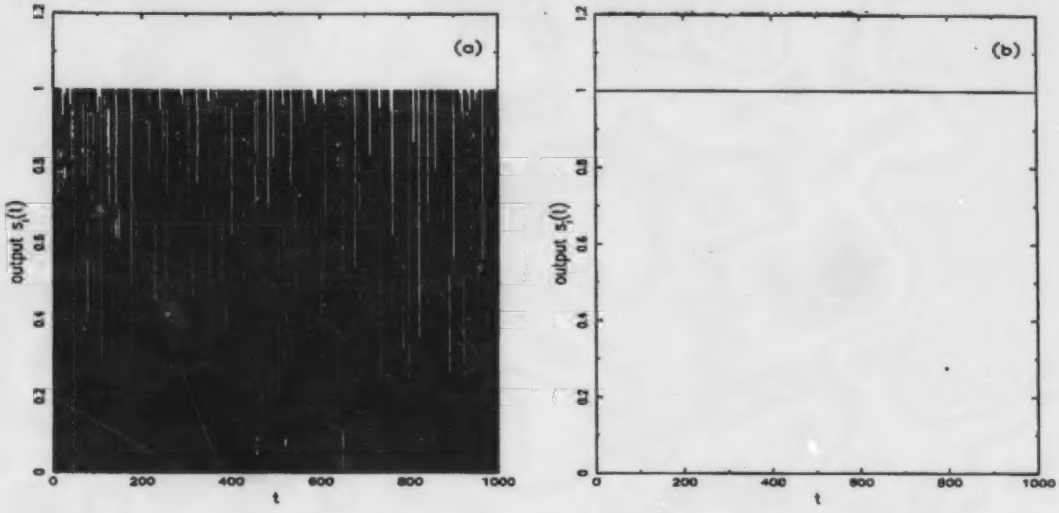

Figure 2.2: The output $s_i(t)$ vs. t in the network with $N = 16, p = 4, k_f = 0.2, a_i = 1., \epsilon = 0.015$ and (a) $k_r = 0.9, \alpha = 5.0$; (b) $k_r = 0.1$, $\alpha = 0.5$. The dynamics in (a) ((b)) is chaotic (nonchaotic).

set of the value of the parameters is $k_f = 0.2, k_r = 0.9, \alpha = 5.0, a_i = 1.0$ and $\epsilon = 0.015$. There are of course regions where the dynamic behavior is a mixture of chaotic and nonchaotic output.

With this model, people have discussed associative memory in chaotic systems and find practical applications that will be discussed below.

2.3 The Associative Memory

As discussed in the introduction, the associative memory in chaotic systems is unclear. On one hand, the dynamics is not convergent. If we insist on the usual concept of association in the nonchaotic case, we have to use new approaches to "help" the dynamics to find a persistence. On the other hand, if we generalize the classic picture of association, we need to provide enough evidence to explain why such generalization is reasonable. For example, if

a state after long enough dynamic evolution is not the pattern near which the system is started, then the usual recognition is failed. But if the state is one that orbits or moves around patterns, can we in fact say more than just "it has failed"? This idea requires generalizing the picture of associative memory (pattern recognition) and it leads us into concepts such as the "cycle of memory" and "random link of memory". In this report, we discuss these two methods. The latter method will be discussed right now, leaving the former for chapter 6.

Adachi, Ikeguchi and Nagashima's groups used the model to explore pattern recognition [13] – [17]. Let us store 4 patterns in the net. The synapses are determined by Eq. (2.10) with $p = 4$. The main result is that the network first presents a nonperiodic behavior in a transient phase, after that the network has in most cases a periodic behavior. The nonperiodic character in the transient phase is described as "memory searching", which means that the network moves randomly around the stored patterns. The period of the periodic trajectory reached after the transient phase is related to the initial patterns. If we check the retrieval rate in the sense of the conventional picture, i.e. the network converges to a stored pattern, the rate is low. However, the meaning of the result, according to [13, 14, 15], is that the memory searching process can be viewed as having the character of an associative memory. Table 2.1 gives the retrieval rate of the four patterns out of 5000 iterations in the transient phase for four different initial conditions denoted by init1, init2, init3 and init4. The parameter is set to be $\alpha = 10, k_f = 0.2, k_r = 0.9$ and $a_i = 2$. The rate is low, if we insist on the exact retrieval of the patterns. After the transient duration, the network evolves periodically. Table 2.2 shows the periods after the transient phases for different initial conditions whose hamming distance h with the stored patterns are 0, 4 and 8, respectively. The numbers in the brackets are the transient durations. The parameters are the same as those used in Table 2.1 In the two nonperiodic cases, the Lyapunov exponents become positive as time increases. The transient time is dependent also on the initial conditions.

The influence of the updating method on the retrieval rate is also discussed. The recurrent relations of (2.5) can be updated sequentially or in parallel. The above discussions are based on parallel updatings. In [13], a "Winner Faster" approach is discussed: The neurons are updated according

Table 2.1: The retrieval rate out of 5000 iterations in the transient phase. $N = 100, \alpha = 10.0, k_f = 0.2, k_r = 0.9$ and $a_i = 2.0, i = 1, \ldots, N$

	init1	init2	init3	init4
ξ_1	27	63	64	14
ξ_2	48	68	61	66
ξ_3	31	21	25	19
ξ_4	21	13	11	28
Total	127	165	161	127

Table 2.2: The periods after the transient phase

	$h = 0$	$h = 4$	$h = 8$
ξ_1	9378	nonperiodic	20 (1000)
ξ_2	9378	20(9000)	20(10000)
ξ_3	20 (21000)	9378	20 (6000)
ξ_4	nonperiodic	9378	20 (4000)

to the values of $\eta_i(t)$, i.e. the neuron with the largest $\eta_i(t)$ is updated first, then the neuron with next to the largest $\eta_i(t)$, and so on. Such sequential updating, according to [13], gives some improvement of the associative property.

These studies demonstrate the existence of a searching process, i.e. the dynamics wander throughout the whole phase space and do not get trapped. Therefore, all patterns could be reached and none of them will be lost or forgotten. Although some people [13, 14, 15] believe that this process already means that the network possesses association, how to find a "definite" pattern is still a problem. This is so, because this dynamics, even when when starting a stimulus from within a neighbourhood of a pattern, will drive the initial state away form the pattern. To consider this problem further, we turn to the external term that was ignored before. If the network has communication with external world, we have to consider the contribution from $\phi(t)$. According to Skarda and Freeman [1], when a chaotic network is stimulated by a known input, the network is constrained to a limit cycle of attractors; when stimulated by a unknown input, the network evolves to a high level of chaotic state that is believed to be critical for the learning of new patterns. This hypothesis is demonstrated in this model. When $\phi(t)$ is set to one of the known patterns, the hamming distance (showing the retrieval behavior) between the output $s(t)$ and the known pattern shows a periodic behavior with the period 114, while the hamming distance between the $s(t)$ and an unknown pattern has a much larger period 5120.

Another discussion related to the association is the clustering. Nagashima *et al* discuss the clustering property of this model [16, 17]. The neurons can be divided into clusters (groups) in terms of their output, the neurons in a same cluster have almost the same responses on a stimulus. The neurons in the same group switch synchronously during the evolution. The patterns are retrieved in a synchronous manner. The association based on the clustering and switching of the neurons are discussed in [16, 17] in detail and it can be considered as another variant of associative memory.

In addition, the model was also applied successfully to practice, Travel Salesman Problem (TSP) [25] and speech recognition [26]. The TSP is to find the shortest total distance travelled for a salesman who must visit a large number of cities that are distributed randomly. This problem has attracted

attention since Hopfield and Tank [27] suggested the solution with neural network. Using the model discussed above, Hasegawa *et al* find a good solution of TSP. They argue that the Chaos in this model plays an essential role to avoid the local minimum. Another application is the speech recognition [26], where the model was used to recognize speech and better results were obtained. However, the synapses used there are no more the Hebbian rule, but steepest descent algorithm. The existence of chaos needs to be checked further.

In short, there is still a lack of valid reasons to believe that the model has associative memory in the sense of finding a *definite* pattern. This is again related to the problem of how one should describe *reasonably* the character of association in the chaotic system, where the conventional concept is no longer valid. The main idea discussed in this chapter is that association must be defined as a cycle of states visiting all possible patterns, not the single pattern itself from the neighbourhood of which the dynamics is begun. A more easily acceptable argument will be discussed in chapter 4 with the same model, based on parameter control. The significance of the discussions in this chapter may be the process of memory searching, as claimed by the contributors.

Chapter 3

Memory Searching in a Network with Synapse Reduction

Nara and Davis *et al* study another memory searching process and association based on the idea of cycle of memories. Chaos is introduced by cutting off a number of connections among the neurons. For some cutoff ratios, the model generates states that visit the basins obtained in fully connected model with definite probability. Such probability can be understood as a criterion for judging the association.

3.1 The Model and Chaos

The model is a spin-glass-like model and the synapses take the Hebbian form. The state of N neurons at time t in this model, $s_i(t) = \pm 1$, evolves with the dynamics

$$s_i(t+1) = \theta(\sum_j^N J_{ij} s_j(t)), \tag{3.1}$$

where, θ is the step function and J_{ij} are the synapses defined by

$$J = \sum_{\mu=1}^{L} \sum_{\lambda=1}^{M} \xi_\mu^{\lambda+1} \otimes \xi_\mu^{\lambda} \tag{3.2}$$

with $\otimes$ being the direct product of patterns denoted by vectors ξ_μ^λ (each pattern is a vector with N components). L and M are the number of cycles and the period of each cycle, respectively, namely $\xi_\mu^{M+1} = \xi_\mu^1$.

Chaos is introduced by removing some connections in the network. Before cutoff, one neuron is connected with $N-1$ neurons through synapses. If a number, r, of them is cut, the network is diluted in the sense of having only $N-1-r$ synapses per neuron. The dynamic behavior depends on r. Fig.

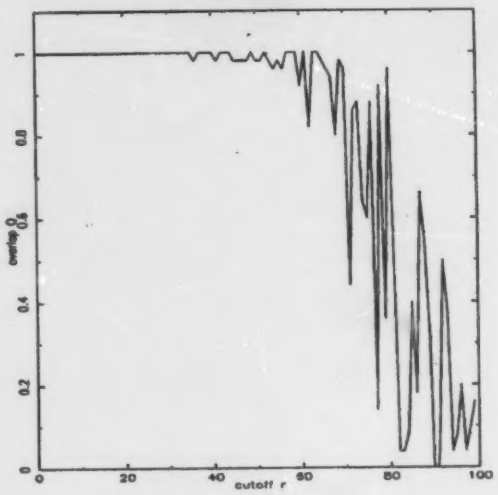

Figure 3.1: The overlap Q vs. the cutoff r for $N = 100$ and 10 patterns.

3.1 shows the overlap,

$$Q = \sum_i^N s_i(t)(\xi_1^1)_i, \tag{3.3}$$

between the state $s(t)$ and a pattern, e.g. ξ_1^1 (whose components are denoted by $(\xi_1^1)_i$), versus the cutoff parameter r after a long time evolution. $s(t)$ is started from ξ_1^1. The overlap becomes chaotic, when r reaches a critical value. However, the chaotic property has not been verified by calculating the Lyapunov exponent or correlation function. A similar approach has been introduced into the Gardner model, where the existence of chaos is confirmed

by calculating the auto-correlation function, which will be discussed in detail in section 5.2.

3.2 Memory Searching

When the cutoff ratio is small, this model has associative memory. A state in the basin of a pattern will be driven to the pattern by the dynamics. When the reduction is increased, an initial state will not converge to a pattern, or to a static state. It wanders in the phase space. Nara and Davis *et al* found that an intial state will visit the basins of *all* patterns. The phase space is first partitioned into basins defined at full connection and the complement of those basins. Although the dynamics makes the state wander in the space, the state visits all basins in a cycle: the visiting probability is time-invariant, except depending on the initial condition. Fig. 3.2(a) and (b) show the number of visits to the basins of patterns $1 \sim 10$ after 1000 and 2000 steps, respectively. The parameter is defined in the caption. Each basin has a probability of visitation and it is almost time independent. If a slightly different initial state is used, the distribution of the probability is much different. Fig. 3.2(c) gives the distribution for an initial state that has only one bit difference from the initial state used in (a) and (b). The very different distribution due to slightly different initial conditions indicates chaotic behavior. However, again it is necessary to take into account the Lyapunov exponents to judge the existence of chaos.

The process of visiting all basins is discussed in [20, 21] as memory searching. After defining a feature parameter, the pattern with a certain value of the parameter can be searched by using the above dynamic process. In addition, the hierarchical intermittence of the chaotic states and possible bifurcation behavior are also discussed in [20, 21].

As in the preceding chapter, it is still not clear, how to introduce associative memory in this model. Although some related discussion is implied in this model, there is lacking a concrete description of the association. Only the process of "memory searching" is emphasized in the Nara *et al* model as in Aihara *et al*'s work. In both models, the cycles of states that visit *all* patterns or basins of patterns are observed, which can be, according to Amari [28], considered as persistence of patterns. However, such cycles depend greatly

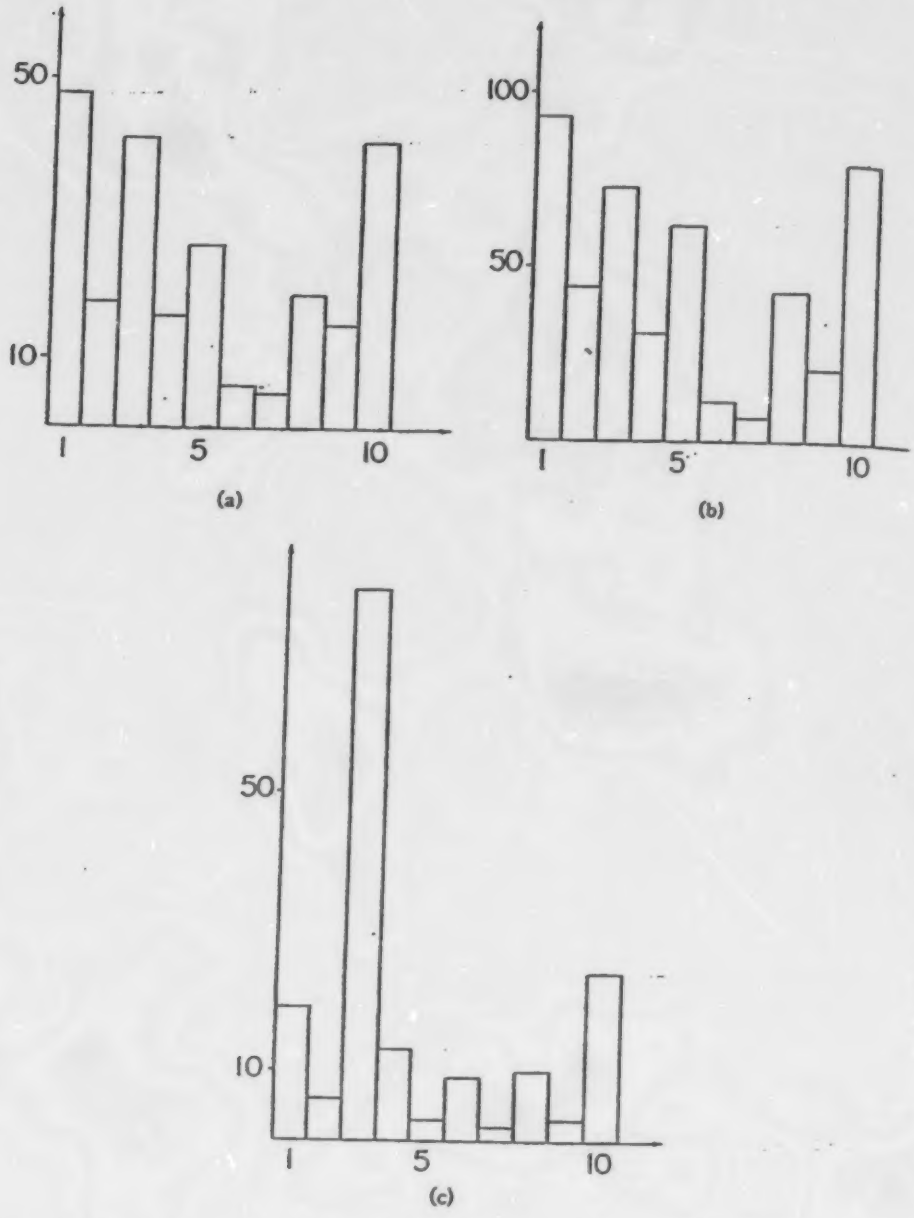

Figure 3.2: The number of visits to the basins of patterns $1 \sim 10$. The same initial inputs are used in (a) and (b), while it is slight different in (c). The distribution in (a) with 1000 iterations is very close to that in (b) with 2000 iterations. In (c) 1000 iterations is used. $N = 400, r = 6, L = 10$ and $M = 3$. (from [21]).

on the initial conditions. A slight deviation of the initial condition will lead to a very different dynamic state, so we still don't have a good idea of how to describe the association as we have done in the nonchaotic models. The following chapters give other methods that are closer to those we are familiar with in the conventional networks.

Chapter 4

Dynamic Link of Memory

The associative property in Tsuda's model is discussed more clearly [18]. In his *chaotic model* Tsuda has shown (1) if an input is similar to a known pattern, the dynamics will find the pattern with *deterministic* law. Depending on coupling parameters, the output is either the pattern, or is confined in the basin of the pattern, presenting local chaotic movement; (2) if an input is very much different form the known patterns, the output exhibits the behavior of memory search that is similar to the discussions in chapter 2 and 3; (3) the new pattern can be learned and such learning can destroy the old memory, which is implied by *Skarda* and Amit *et al.* Therefore, chaos in this model plays a role to find the memorized items, i.e. a dynamic link of the memory.

4.1 A Model Composed of Two Modules

The model is composed of two modules (denoted by M_a with $a = 1, 2$) that have different functions and the two parts are coupled together. Both M_1 and M_2 have number of m neurons denoted by $i_1 = 1, \ldots, m$ and $i_2 = 1, \ldots, m$, respectively. The synapses, $J_{i_a j_a}$, connecting the neurons within M_a, are updated as

$$J_{i_a j_a}(t+1) = J_{i_a j_a}(t) + \epsilon s_{i_a}(t) s_{j_a}(t), \quad a = 1, 2, \tag{4.1}$$

while the synapses, $J_{i_a j_b}$, connecting the neurons in M_a and M_b with $a \neq b$, take the form of

$$J_{i_a j_b}(t+1) = J_{i_a j_b}(t) + \epsilon' s_{i_a}(t) s_{j_b}(t), \quad a \neq b, \quad , a, b = 1, 2, \tag{4.2}$$

where ϵ and ϵ' are small constants. The two kinds of synapses are set initially by using the Hebbian rule

$$J_{ij}(0) = \sum_{\mu}^{p} \xi_{\mu}^{i} \xi_{\mu}^{j}, \quad i, j \in M_1 \cup M_2 \tag{4.3}$$

with ξ_μ the pattern stored in the net. The dynamics in the net has the form

$$s_{i_a}(t+1) = f^{p_a}\left(\sum_j J_{i_a j} s_j(t) + d_{i_a} r_{i_a}(t) - \delta_{a,2} \phi_{i_a}(t) \right), \tag{4.4}$$

where, d_{i_a} is a constant, $r_{i_a}(t)$ evolving as

$$r_{i_a}(t+1) = f^{p_r}(\sum_{j_a \in M_a} e_{j_a} s_{j_a}(t)) \tag{4.5}$$

with e_{j_a} a positive constant. $s_{i_a}(t)$ and $r_{i_a}(t+1)$ are independent stochastic variables described by a probability function f^z

$$y(t+1) = f^z(x(t)) = \begin{cases} 1 & \text{if } x(t) > 0, \\ -1 & \text{with probability } z, \text{ if } x(t) < 0, \\ y(t) & \text{with probability } 1-z, \text{ if } x(t) < 0. \end{cases} \tag{4.6}$$

$\delta_{a,2} = 0$ or 1 for $a = 1$ or 2, respectively. The negative feedback term, $\phi_{i_a}(t) = \sum_{j_2 \in M_2}^{m} J_{i_a j_2} s_{j_2}(t)$, satisfies the condition $\phi_{i_a}(t+1) = \phi_{i_a}(t)$, i.e. it goes to work when a steady state is reached.

The first term in Eq. (4.4) is the usual contribution from the neurons, the second term introduces a stochastic term that is generated by a feedback from the output $s(t)$ through strength e_j and the third term (negative feedback) exists only in the module 2, which makes the neurons out of stable states. In his paper Tsuda has given some biological backgrounds to establish the model. In short, the first term is for the memory storage and make the dynamics convergent. The influences from other terms make the system be in a nonequilibrium state. The competition of these two forces set the system in stable, or temporarily unstable states, depending on the parameters.

4.2 Association and Dynamic Link

Chaos was shown to exist by calculating the Lyapunov exponent. The model was first checked by inputing the learned patterns. All the inputs were retrieved by the model and this provides the difference between the results discussed in chapter 2, where the learned inputs lead to a chaotic wandering in the phase space.

Then stimulus that is somewhat different from a learned pattern is input to the net. Three kinds of outputs are obtained: (1) when the coupling parameters are large, the output is the input, no matter what the input is; (2) when the couplings are somewhat weak, the output is in the basin of the pattern when the input is similar to the pattern or (3) the output exhibits chaotic searching in the phase space when the input is very different from the learned patterns. The last point bears the comparison with the discussion in chapter 2 and 3. The retrieval is related to the coupling, the stronger the coupling, the more the the chaotic behavior is restricted locally around the pattern.

In addition, Tsuda has also discussed the learning process by setting ϵ, $\epsilon' \neq 0$. He finds in some cases that if a new pattern is learned by the net, the old memory will be lost. Therefore, the Skarda and Freeman's hypothesis about the chaos as a "novel filter" was shown. Chaos works like a trigger making a system move form the old state to a new state. The similar mechanism of generation of association based on a transition through, fixed point $\rightarrow$ chaotic attractor $\rightarrow$ new fixed points, was also shown in [19].

In short, Tsuda has shown that association exists in his chaotic model and chaos is useful to learn new items. The association discussed here is more clear than those in chapter 2 and 3, since a *definite* pattern can be retrieved from the input near the pattern, although the dynamic is chaotic. The model tells us that there is a deterministic mechanism (retrieve a pattern with partial information on it), even the dynamics is chaotic. Of course, the mechanism is realized by defining carefully the model.

The drawback of the model is its complexity. Because of the complicated coupling and design, it is not easy to apply the model to practice.

Chapter 5

Chaos Control

As we have mentioned many times, description and definition of association in models with chaos is a nontrivial task. In the preceding chapters, we discussed association within a dynamic picture: the states will not be fixed on a stable point but rather it will move *around* the patterns (fixed points). The period or the visiting probability of the cycles of the states provides us information about registering association. However, it is difficult to connect such arguments with the association that we are used to in the conventional sense. In this chapter, we focus on the conventional idea of pattern recognition: the real state should approach the *desired pattern* if the pattern is recognized. In order to perform the task, we resort to controlling the chaos present in the evolution. There are many methods for controlling chaos [29] - [33]. In the following we introduce two typical ways: parameter control and feedback control.

5.1 Parameter Control

With the same model discussed in chapter 2, Kushibe, Liu and Ohtsubo utilized parameter control to implement associative memory [22]. As we will see, this idea is acceptable in so far as it is related closely to the conventional picture of pattern recognition that we are familar with. Let us first give a brief description of the model. According to the original model [12] and simplifying the Eqs. (2.5) - (2.8) by setting $k_r = k_f = k$, the output of

neuron i at time t can be written

$$s_i(t+1) = f(y_i(t+1)) \tag{5.1}$$

with

$$y_i(t+1) = ky_i(t) + \sum_{j=1}^{M} J_{ij}s_j(t) - \alpha s_i(t) + a_i \tag{5.2}$$

and

$$f(z) = \tanh(\frac{z}{\epsilon}), \tag{5.3}$$

where ϵ is a positive constant. The synapses J_{ij} obey the Hebbian rule

$$J_{ij} = \frac{1}{p}\sum_{\mu=1}^{p} \xi_i^\mu \xi_j^\mu, \tag{5.4}$$

where, ξ_i^μ is the i-th element of pattern μ and p is the number of total patterns.

The largest Lyapunov exponent has been calculated, in order to confirm the presence of chaos. The shaded region in Fig. 5.1 corresponds to positive values of the Lyapunov exponent, while the white area indicates negative values. Therefore, the model is chaotic in a wide region of the parameters. When the parameters k, α and a_i in Eq. (5.2) are zero, the model reduces to the Hopfield model. Therefore, if we can control the parameters in the dynamics, so that they become small we can use this model as an associative memory. At the same time, the local minima can be avoided by the chaotic property. The methods of controlling parameters are not unique (see e.g. [29] – [33]). Kushibe *et al* use the following:

$$k(t) = [1 - c^\mu(t)]k(0), \tag{5.5}$$

$$\alpha(t) = [1 - c^\mu(t)]\alpha(0), \tag{5.6}$$

and

$$a_i(t) = [1 - c^\mu(t)]a_i(0), \tag{5.7}$$

where, $k(0), \alpha(0)$ and $a_i(0)$ are the initial values of the parameter k, α and a_i, respectively. $c^\mu(t)$ is the control signal for pattern μ, the target pattern.

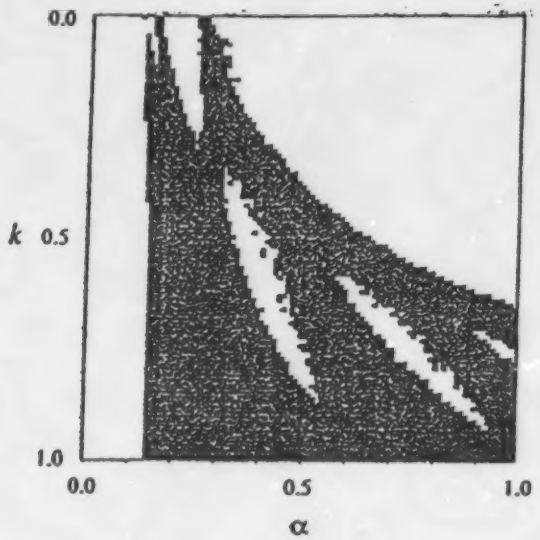

Figure 5.1: The Lyapunov exponent diagram versus k and α with $M = 100, \epsilon = 0.02$ and $a_i = 0.2$ (from [22]).

The control signal is connected with the overlap between the state $s_i(t)$ at time t and the pattern ξ^μ by

$$c^\mu(t) = \frac{1}{N_s} \sum_{i=1}^{N_s} g_i^\mu(t) \tag{5.8}$$

with

$$g_i^\mu(t) = \mathrm{sgn}[s_i(t)\xi_i^\mu] \exp\left(\left|\frac{s_i(t)}{\xi_i^\mu}\right| - 1\right), \tag{5.9}$$

where, N_s is the sampling number chosen randomly out of the N neurons. Recalling that $s_i(t)$ takes values between -1 and $+1$, $g_i^\mu(t)$ is therefore within -1 and $+1$ and it is $+1(-1)$ if it is completely the (inverse) pattern. As the state $s_i(t)$ is approaching the pattern ξ_μ, $g_i^\mu(t)$ is approaching 1 and from the control equations of (5.5) – (5.7), the parameters approach zero. The model is therefore reduced to be the Hopfield model. The analysis of the dynamics (section 2.2) and the Lyapunov exponent (Fig. 5.1) reveals the important role of the parameter α: on the one hand, the larger the parameter, the more flexible (chaotic) the network and the state will not be trapped in any local

minimum; on the other hand, too large α will make the state wander in the phase space and it will not converge to a pattern. In order to overcome this dilemma, an extra learning algorithm is imposed on Eq. (5.6) by modeling the external input a_i

$$\text{if} \quad c^{\mu}(t) \geq \gamma, \text{then} \quad a_i = [1 - c^{\mu}(t)][a_i(0) + \Delta\xi_i^{\mu}], \tag{5.10}$$

where, γ is a threshold between 0 and 1, Δ is a reinforcement constant. The Eq. (5.2) is now

$$y_i(t+1) = ky_i(t) + \sum_{j=1}^{M} J_{ij}s_j(t) - [\alpha s_i(t) - \Delta(t)\xi_i^{\mu}] + a_i, \tag{5.11}$$

where $\Delta(t) = \Delta[1 - c^{\mu}(t)]$. Thus, if the state $s_i(t)$ is approaching the pattern ξ_{μ} ($c^{\mu}(t) \geq \gamma$), the chaos becomes weak and the dynamics will find the pattern easily; if the state is far from the pattern ($c^{\mu}(t) < \gamma$), we adapt the old equation (5.2). The extra learning rule is aimed at helping the states near a pattern reach that pattern but does not influence the states that are far away from the pattern.

Fig. 5.2 shows the associative property of the network when 6 patterns are stored. It gives the relation between the retrieval rate S_r and the sampling number for various reinforcement. The sampling neurons are initialized to be the elements of known patterns, while the neurons outside the sampling are given initially in a random manner. The retrieval rate is defined as the ratio of the number of successful retrieval of a pattern to the total number of trials. From the picture we see that reinforcement does play a positive role in giving a better associative performance. In [22], the relationships between the searching time, the retrieval rate, the parameter $k(0)$, $\alpha(0)$ and the reinforcement Δ are also discussed. The results show that the model demonstrates associative ability when chaos is controlled. The association discussed here is the conventional concept accepted in nonchaotic models. The reinforcement process improved associative memory. In short, chaos is used and suppressed during the association. One uses chaos to avoid local minima and suppresses it to find a pattern. The closer the state gets to the pattern, the weaker the chaos gets. When the state coincides with the pattern, the chaos is zero. The model reduces completely to the Hopfield model. In order to recognize patterns, the model must reduce to the Hopfield model. At the same time, it has an advantage over the Hopfield in that chaos,

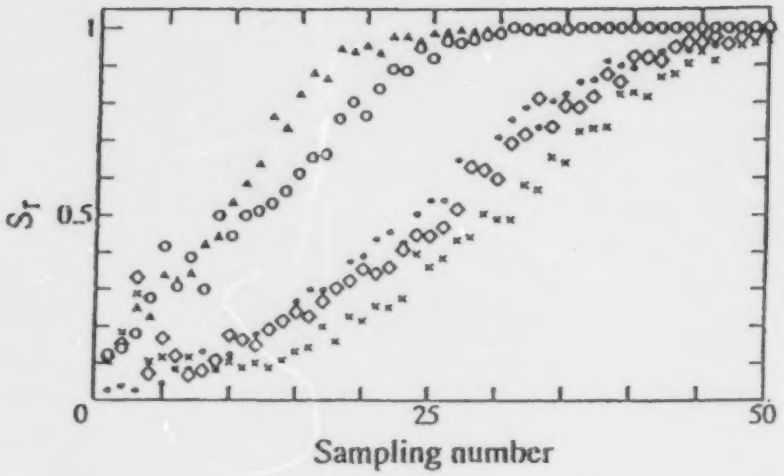

Figure 5.2: The retrieval rate S_r versus the number of sampling neurons N_s for $\Delta = 0$ (crosses), 0.1 (diamonds), 0.5 (triangles) and 0.9 (circles). Filled circles are the results of the Hopfield model. The parameters are $N = 100, k(0) = 0.9, \alpha(0) = 0.6$ and $a_i(0) = 0.2$. The Lyapunov exponent is 0.3911 (from [22]).

when present, allows the dynamics to escape local minima. Therefore, the results shown in [22] are better than those in the Hopfield model. This advantage of chaos has also been pointed out in section 2.3, in regards to the Travelling Salesman Problem and to speech recognition. Association in these examples is reasonable and easy to understand.

5.2 Trajectory Control

In what follows, we describe the techniques that we have used in pattern recognition with chaotic systems. First we discuss the method based on chaos control. We have also used the idea of chaos control in other models. Some of the work can be found in [23, 41]. Pattern recognition using synchronization of chaotic systems does not require the patterns to be associated with any fixed points. We illustrate the method by considering two different chaotic neural networks.

5.2.1 The Gardner Model

Here we address the issue of pattern recognition using a model chaotic network obtained from the well known Gardner model [8].

Chapter 2 and 3 have shown that in the presence of chaos the dynamics wanders among the learned patterns. Chapter 4 has presented a model for a dynamic link of memory in nonequilibrium neural network. It is not clear how the correct memory is retrieved from the wandering dynamics. In this section we demonstrate numerically that the history of trajectories contain information about patterns and we propose an approach to extract this hidden information. Our ansatz is based on synchronization of chaotic systems using the feedback method. In what follows, first we introduce chaos in the Gardner model and then discuss the process of pattern recognition.

In the Gardner model [8], N neurons are *fully* connected to one another through the synapses J_{ij}, $i, j = 1, \ldots, N$. The state, $S_i(t)$, of the i-th neuron at time t is the spin variable $S_i(t) = \pm 1$. In order to store p patterns,

$\xi_i^\mu = \pm 1, \mu = 1, \ldots, p$, the synapses are trained so that the condition

$$\frac{1}{\sqrt{N}} \xi_i^\mu \sum_{j \neq i} J_{ij} \xi_j^\mu > K \geq 0, \quad \mu = 1, \ldots, p \tag{5.12}$$

is satisfied. This model can work as an associate memory: it is capable of remembering maximum $2N$ patterns and retrieving these patterns, based on partial information about them [8, 9, 10]. The patterns coded in synapses are the fixed points under the dynamics of the system. The larger is the parameter K, the larger are the basins of fixed points. Details of the Gardner model is available in the literatures.

Now we introduce chaos into this model by cutting randomly a number of synapses. Chaos appears if a neuron is connected to fewer than the maximum of $N-1$ neurons in a network of N neurons. The number of missing synapses, chosen randomly, is represented as a fraction c of the total $N-1$ synapses for any neuron. This way of introducing chaos could be significant, since a network with partially connected neurons has a similarity with the human brain in which the neurons are not fully connected [34] and the loss of synaptic connections in the human brain may also occur because of brain damage [3]. In the Gardner model, the cutting off of synapses (CUTS) has been discussed earlier as a diluted approach [35], where the effect of the dilution on the retrieval of patterns corresponding to the fixed points were considered. However, the existing work concentrates exclusively on the concept of pattern recognition being persistent under dynamics without any analysis of its relation to chaos. We first show that chaos is indeed introduced in this way. The usual way to identify chaos in a system is to calculate the Lyapunov exponents or auto-correlation function [36, 37]. We have used both of these techniques to examine the presence of chaos. Here we present only the results of the calculation of auto-correlation function which is defined as

$$C(\tau) = \lim_{T \to \infty} \frac{1}{T} \sum_{t=0}^{T} S_i(t) S_i(t+\tau) \tag{5.13}$$

where $S_i(t)$ is the state of neuron i at time t. $C(\tau)$ is independent of the choice of i. The criterion of the auto-correlation function states that if $C(\tau)$ is a decreasing function of τ, the system is chaotic. The speed at which $C(\tau)$ decreases is related to the rate at which the systems turns chaotic. The dependence of $C(\tau)$ on τ is presented in Fig. 5.3. The figure shows that when

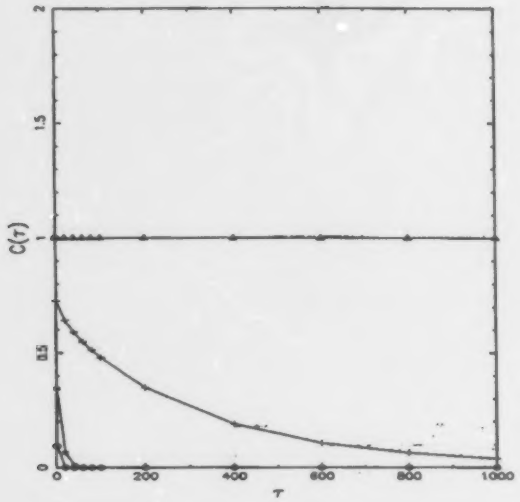

Figure 5.3: The auto-correlation function of the Gardner model at $K = 2$ for $c = 0$ (triangles), 0.5 (crosses), 0.6 (stars) and 0.8 (circles). Points are linked for a better perception.

$c = 0$, $C(\tau)$ remains a constant while for $c > 0$, $C(\tau)$ decreases from a finite value to zero. For larger values of c, $C(\tau)$ goes to zero faster implying a rapid growth of chaotic behavior. It should be pointed out that $C(\tau)$ is also dependent on K. For smaller K (not shown), even a small c will produce chaos.

When chaos is introduced, the network dynamics will not drive an initial state to a fixed point (pattern) as in the nonchaotic model without CUTS. The condition (5.12) is violated and patterns ξ^μ are no more the fixed points. In fact, a chaotic system has a large number of unstable period orbits (UPO) [32, 33]. Although the dynamics does not drive an initial state to a fixed point, patterns may still be recognized. The synapses are trained before the CUTS and therefore, we believe that some information about the patterns may still be present in the remaining J_{ij}. Before a fully chaotic behavior sets in, it may be possible to use the network for pattern identification. We are pointing here to short-term memories. The problem now is to extract this information, if there is any. To this end, we turn to synchronization of identi-

cal chaotic systems. The approach to synchronization of chaotic trajectories is not unique [32, 33]. Here we utilize the feedback method proposed in [32] and select the trajectory stimulated initially by a pattern (e.g. ξ^1) as desired orbit and synchronize other chaotic trajectories to this orbit. We try to get useful information during the synchronization.

We denote the desired trajectory as $\zeta^1(t)$ with $\zeta^1(0) = \xi^1$ and a trajectory to be synchronized as $s(t)$ which is near ξ^1 at $t = 0$. The relation between $\zeta^1(t)$ and $s(t)$ is measured by a dynamic distance

$$d^1(t) = \left\langle \frac{1}{N} \sum_{i=1}^{N} (s_i(t) - \zeta_i^1(t))^2 \right\rangle_\xi , \tag{5.14}$$

where, $\langle \cdots \rangle_\xi$ means the average over random patterns ξ in the Gardner model [7, 8, 9, 10]. $d^1(0)$ gives the initial distance. Although $\zeta^1(t)$ is not ξ^1, it is generated by ξ^1 and in a deterministic chaotic system it is unique. Therefore, we "transfer" ξ^1 to $\zeta^1(t)$. If $d^1(t)$ goes to zero (or a small value), ξ^1 is retrieved from $s(0)$ containing only partial information of ξ^1. In this chaotic model, the dynamics will drive $s(t)$ away from $\zeta^1(t)$. $d^1(t > 0)$ is larger than $d^1(0)$. But if at every dynamic step we "help" $s(t)$ to go a little bit closer to $\zeta^1(t)$, we may get useful results. This is just the idea of chaos control by feedback. We now turn to chaos control and try to perform recognition in the chaotic system. The feedback is imposed by modifying the trajectory $s(t)$ at time t with

$$s_i(t) \to s_i^f(t) = s_i(t) + k(\zeta_i^1(t) - s_i(t)), \quad i = 1, \ldots, N \tag{5.15}$$

where $0 < k < 1$ is a constant called hereafter as the feedback strength. For $k = 0$, there is no feedback; $k = 1$, the trajectory is modified immediately to be the desired one. The state is $s^f(t)$ after the feedback process is utilized for the next dynamics. Because the feedback is applied directly and at every time step to the output of the system, $s^f(t)$ will be driven to $\zeta^1(t)$ after sufficiently long time depending on k. After the synchronization, the feedback term goes to zero and the distance in Eq. (5.14) vanishes. However, we cannot believe that $d^1(t) = 0$ in this case means that $s(t)$ is recognized, because $s(t)$ may be synchronized in the same way with all other trajectories $\zeta^2, \ldots, \zeta^p$ coming initially from other patterns, $\xi^2, \ldots, \xi^p$, respectively. Therefore, we need another criterion to distinguish between the synchronized orbits.

The criterion that we have used in this work is based on "time of synchronization". The phrase time of synchronization (TOS) is used here in a limited sense. TOS is the time required for the distance between the drive and response systems to become less than a preset value d_0. According to this criterion, TOS tells us which (target) pattern is retrieved. More precisely, we denote $d^\mu(t)$ as the distance by replacing index 1 in Eq. (5.14) with μ. The shortest time

$$t^r = min\{t^1, \ldots, t^p\}, \tag{5.16}$$

where t^μ satisfy

$$d^\mu(t^\mu) < d_0, \quad \mu = 1, \ldots, p, \tag{5.17}$$

determines that pattern r is the target pattern. Here d_0 is a small number. If the shortest time t^r does not exist (the t^μ, $\mu = 1, \ldots, p$ are close to one another before d_0 is reached), then the recognition is failed. In practice we do not have to compare all distances, because after the average in Eq. (5.14) all patterns are equivalent. Below we set $s(0)$ to be related to ξ^1 with the initial distance $d^1(0)$ and compare the distance $d^1(t)$ between $s(t)$ and $\zeta^1(t)$ and that $d^2(t)$ between $s(t)$ and an arbitrary trajectory, see $\zeta^2(t)$ starting from ξ^2. Since the patterns are randomly orthogonal, $d^2(0)$ is 2. In the following, we choose $d_0 = 0.02$.

The approach has been first checked for the fully connected Gardner model ($c = 0$). Fig. 5.4(a) and Fig. 5.4(b) gives the distances $d^1(t)$ (circles) and $d^2(t)$ (crosses) versus t with initial distance $d^1(0) = 1$ for $K = 2$ and for $K = 0.68$, respectively. In Fig. 5.4(a) $d^1(t)$ is always smaller than $d^2(t)$ and this leads to the result that t^1 is smaller than t^2, while in Fig. 5.4(b) $d^1(t)$ and $d^2(t)$ have the same value after certain steps. Therefore, according to our discussion, the network is able to retrieve pattern ξ^1 in Fig. 5.4(a) but not in Fig. 5.4(b). This is in agreement with the well-known results obtained by means of fixed points [8, 9].

Now we turn to the chaotic model. Fig. 5.5 gives the dynamic distance $d^1(t)$ (circles) and $d^2(t)$ (crosses) versus t for $K = 2$, $d^1(0) = 0.6$ and different fractions of CUTS and feedback strengths. In Fig. 5.5(a) with $c = 0.5$, $d^1(t)$ goes to zero much faster than $d^2(t)$ and this implies that t^1 is smaller than t^2 when condition (5.17) is fulfilled. When $c = 0.6$ in Fig. 5.5(b), we cannot find any difference between $d^1(t)$ and $d^2(t)$ after certain steps. This will not lead to different t^1 and t^2 when condition (5.17) is satisfied. Therefore, the pattern is recognized in Fig. 5.5(a) but not in Fig. 5.5(b).

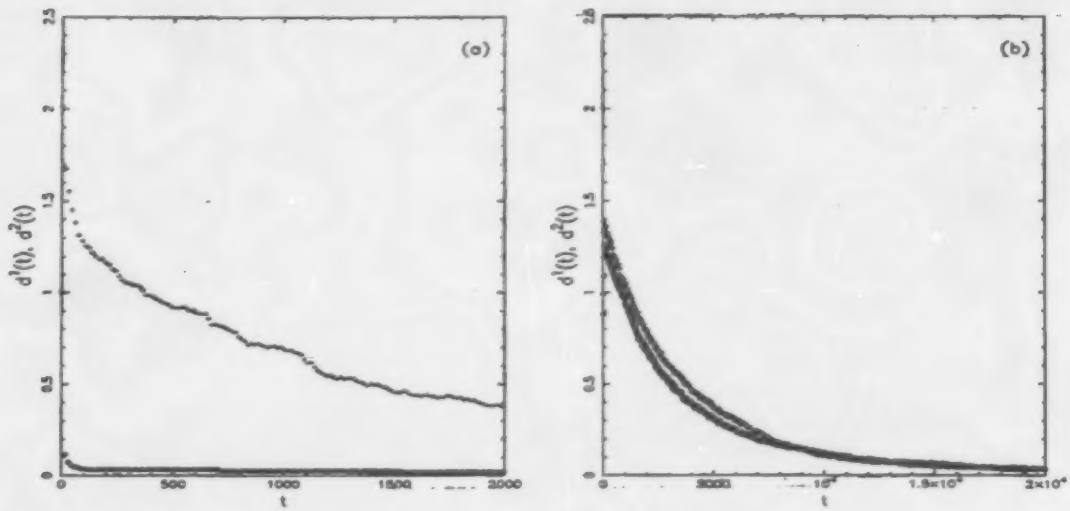

Figure 5.4: The dynamic distances $d^1(t)$ (circles) and $d^2(t)$ (crosses) vs. time t in the fully connected Gardner model ($c = 0$) for $d^1(0) = 1$, (a) $K = 2$, $k = 0.08$; (b) $K = 0.68$, $k = 0.3$. Here k is the feedback strength. The pattern can (not) be recognized in (a) ((b)).

It is natural that when c is small, the structure of the fully connected model will not be affected dramatically. Specially when the fixed points have larger basins (e.g. $K = 2$), the strong convergent tendency persists. When c is large, things change noticeably. When the basins get smaller, even small CUTS may bring chaos. Therefore, the ability of recognition dependents on the parameters K (which determines the basins of the fully connected non-chaotic model), c (which defines the fraction of CUTS) and $d^1(0)$ (which is the initial distance). The critical relation of the parameters has been determined through detailed studies (not given here). It may be pointed out that the qualitative property is not sensitive to the feedback strength k. k determines the time required for satisfying the condition (5.17) and does not affect the order in magnitudes among different t^μ.

Using chaos control and the criterion of time of synchronization, we have shown that the information hidden in the history of evolution can be ex-

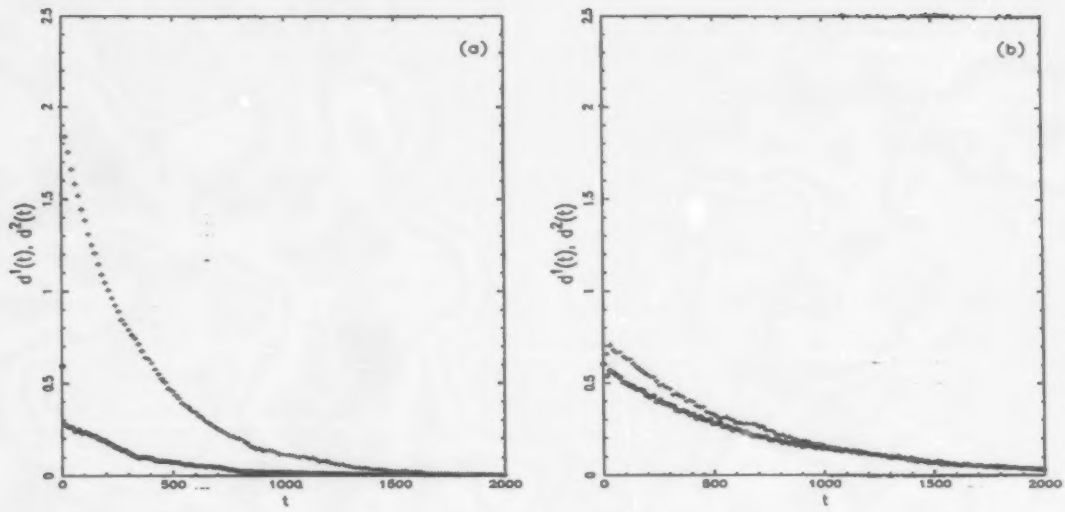

Figure 5.5: The dynamic distances $d^1(t)$ (circles) and $d^2(t)$ (crosses) vs. time t for $K = 2$, $d^1(0) = 0.6$, (a) $c = 0.5$, $k = 0.025$; (b) $c = 0.6$, $k = 0.32$. Here k is the feedback strength. The pattern can (not) be recognized in (a) ((b)).

ploited. When chaos exists, the process of pattern recognition can be performed in this way. We emphasize that this conclusion is not only related to the initial condition, but also to the structure of the network which has embedded the patterns into synapses J_{ij}. This can be verified with starting ζ^1 and ζ^2 by two random states instead of ξ^1 and ξ^2. Fig 5.6 gives the distance $d^1(t)$ between $\zeta^1(t)$ and $s(t)$ and the distance $d^2(t)$ between $\zeta^2(t)$ and $s(t)$. Even for very small initial distance ($d^1(t) = 0.2$) and strong convergent tendency ($K = 2$), $d^1(t)$ and $d^2(t)$ quickly become similar and we cannot treat them the way we did in Fig. 5.5(a). This is not surprizing, because the initial random states are not stored in the J_{ij}. The network "does not know" the random states. The approach of pattern recognition in chaotic networks proposed in this paper can be easily applied to other chaotic systems. It may be interesting to study the effects of gradual and sudden cut off (or regeneration) of synapses on such issues as metastable states [11].

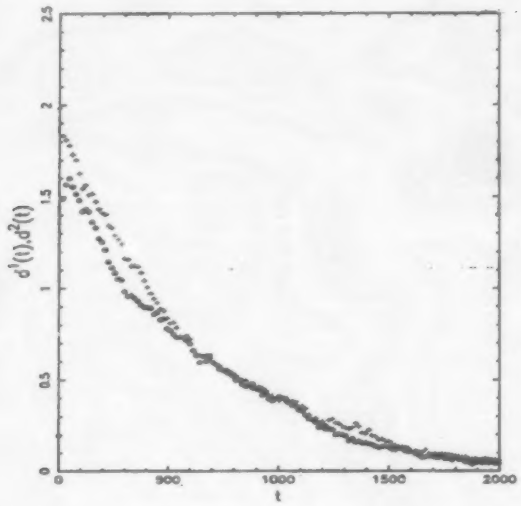

Figure 5.6: The dynamic distances $d^1(t)$ (circles) and $d^2(t)$ (crosses) started from two random states for $K = 2$, $d^1(0) = 0.2$, $c = 0.5$ and $k = 0.023$. The distances are indistinguishable after certain steps.

5.2.2 Quadratic and Cubic Maps and Continuous Hopfield Model

We have used synchronization of chaotic systems for recognition of patterns in various discrete and continuous coupled lattices. As stated above, fixed points are not required for pattern recognition. Using parallel dynamics and the patterns as initial conditions for the lattices, it is shown that a corrupted pattern, reasonably close to its parent, will synchronize with that parent's trajectory under competitive feedback to 'all' known patterns. Also, for one method given, if the corrupted pattern fails to synchronize with its parent it will not synchronize at all, providing a meaningful "I don't know" buffer between patterns. The methods of competitive feedback used are described and the results of recognition trials are given. The coupled map lattices of our investigations have the general form

$$x_{n+1}(i) = f_i(x_n) = f_i(x_n(1), x_n(2), \ldots, x_n(L)) \qquad (5.18)$$

$$y_{n+1}(i) = f_i(y_n) + G_i(x_n, y_n), \tag{5.19}$$

where, n denotes the discrete time and i the spatial coordinate. The vectors $[x_n(1), x_n(2), \ldots, x_n(L)]^T$ and $[y_n(1), y_n(2), \ldots, y_n(L)]^T$ respectively represent the drive and response systems and the G_i are the feedback functions. Here the superscript T denotes transpose.

A drive system represents a pattern and a response system represents some corrupted pattern. For a given response system, there are as many drive systems as there are patterns. The response system and all the drive systems are run in parallel. Each drive system is initialized with a pattern and the response system is initiated with a corrupted pattern. Recognition occurs, not upon arrival at one of the fixed points, but through synchronization of the vector $[y_n(1), y_n(2), \ldots, y_n(L)]^T$ with one of the drivers. An interesting upshot is that the method can be carried out with a variety of mappings and is not limited to neural network architecture. Below we discuss three different types of feedback functions that we have used in our study:

1. Nonlinear Feedback Functions.

$$G_i(x_n, y_n) = \sum_j^L \frac{\partial f_i(x_n)}{\partial x_n(j)}(x_n(j) - y_n(j)). \tag{5.20}$$

2. Linear feedback functions with distance dependent feedback strength.

$$G_i(x_n, y_n) = \gamma \frac{D(x_n(i), y_n(i)) - D(x_{n-1}(i), y_{n-1}(i))}{D(x_n(i), y_n(i))}(x_n(i) - y_n(i)). \tag{5.21}$$

The function D gives the Euclidean distance between the vectors it operates on. The amplitude of the feedback coefficient depends on the ratio of the distance at time n and the distance at time $n-1$ over the distance at time n. In this way we can guarantee that we diminish the distance at every iteration. γ is a constant taken to be 1.

3. Linear feedback with competitive feedback strength.

$$G_i(x_n, y_n) = \sum_{\mu=1}^{p} \eta(\mu)(x_n(i) - y_n(i)) \tag{5.22}$$

with

$$\eta(\mu) = A \exp[-\sigma(D(x_n(\mu), y_n(\mu)))^2], \tag{5.23}$$

where, $x_n(\mu)$ is the driver vector for pattern μ, p the total number of patterns and σ constant taken to be 0.27. The normalization constant A is calculated at every step along with the coefficients. This approach is competitive since the distribution of η for each pattern provides for competition, the closest pattern contributing the most to the change in the response y_n.

We used the above feedback approaches with the following three systems:

1. The cubic map

$$f_i(x_n) = (1-\epsilon)(\alpha x_n(i))^3 - (\alpha+1)x_n(i)) + \frac{\epsilon}{L}\sum_{j=1}^{L}(\alpha x_n(j))^3 - (\alpha+1)x_n(j)), \tag{5.24}$$

2. The logistic map

$$f_i(x_n) = 4x_n(1 - x_n) \tag{5.25}$$

3. The continuous Hopfield model

$$\frac{dx(i)}{dt} = -x(i) + \sum_{j=1}^{L} J_{ij}x(j). \tag{5.26}$$

In the case of the Hopfield model, the differential equations were solved using a fourth order Runge-Kutta method with the feedback added after every iteration. The number of the patterns was equal to the number of nodes.

The tests were carried out by choosing at random one of the known patterns. The driver is initialized with the chosen pattern. The response was initiated with one of the patterns corrupted by some amount of noise. The relation between the driver and the response systems is measured by the hamming distance, a dot product between the two vectors. The trial was recorded as a success if the response vector synchronized with the driver, as a failure if it synchronizes with the wrong driver (a lattice corresponding to another pattern), and as inconclusive if it did not synchronize at all within the maximum number of iterations. The percentages of successful, failed,

and inconclusive trials are given versus the initial dot product of the test pattern with its driver.

The results are given in the following tables.

Table 5.1: Retrieval in the system of cubic map with direct feedback

Initial dist.	Success	Failure	Inconcl.	Total	Success(%)
0.0 - 0.1	0	8	0	8	0.0
0.1 - 0.2	85	180	0	265	32.1
0.2 - 0.3	994	68	0	1062	93.6
0.3 - 0.4	1321	0	0	1321	100.0
0.9 - 1.0	1351	0	0	1351	100.0

Table 5.2: Retrieval in the system of logistic with direct feedback

Initial dist.	Success	Failure	Inconcl.	Total	Success(%)
0.3 - 0.4	0	0	0	0	0.0
0.4 - 0.5	0	0	0	0	0.0
0.5 - 0.6	0	1	0	1	0.0
0.6 - 0.7	6	1983	0	1989	0.3
0.7 - 0.8	680	2644	0	3324	20.5
0.8 - 0.9	1185	1690	0	2875	41.2
0.9 - 1.0	2811	0	0	2811	100.0

It is important to note that these trials are carried out for a loading of 1, that is 256 nodes with 256 patterns. This is especially to be noted in the case of the Hopfield network.

It can be seen that the best results are obtained for the globally coupled cubic map with direct feedback. 100% recovery was achieved for initial overlaps as small as 0.3, and greater than 90% in the range 0.2 - 0.3. Another interesting result is that with the distributed coefficients method, by tuning the distribution parameter sigma, a buffer region can be created where fewer

Table 5.3: Retrieval in the system of cubic map with derivative feedback

Initial dist.	Success	Failure	Inconcl.	Total	Success(%)
0.0 - 0.1	4	1423	0	1427	0.3
0.1 - 0.2	72	1447	0	1519	4.7
0.2 - 0.3	132	1250	0	1382	9.6
0.3 - 0.4	24	1304	0	1328	1.7
0.4 - 0.5	127	1212	0	1339	9.5
0.5 - 0.6	581	730	0	1311	44.3
0.6 - 0.7	1112	227	0	1339	83.0
0.7 - 0.8	1434	26	0	1460	98.2
0.8 - 0.9	1544	0	0	1544	100.0
0.9 - 1.0	1351	0	0	1351	100.0

Table 5.4: Retrieval in the system of logistic map with derivative feedback

Initial dist.	Success	Failure	Inconcl.	Total	Success(%)
0.4 - 0.5	0	0	0	0	0.0
0.5 - 0.6	0	4	0	4	0.0
0.6 - 0.7	2	1783	0	1785	0.1
0.7 - 0.8	2	1571	0	1573	0.1
0.8 - 0.9	967	637	0	1604	60.3
0.9 - 1.0	1034	0	0	1034	100.0

Table 5.5: Retrieval in the system of cubic map with competitive feedback

Initial dist.	Success	Failure	Inconcl.	Total	Success(%)	
0.0 - 0.1	0	1099	1308	2407	0.0	45.7
0.1 - 0.2	0	692	847	1539	0.0	45.0
0.2 - 0.3	0	644	738	1382	0.0	46.6
0.3 - 0.4	21	564	743	1328	1.6	42.5
0.4 - 0.5	196	433	710	1339	14.6	32.3
0.5 - 0.6	586	110	615	1311	44.7	8.4
0.6 - 0.7	690	7	642	1339	51.5	0.5
0.7 - 0.8	779	0	681	1460	53.4	0.0
0.8 - 0.9	898	0	646	1544	58.2	0.0
0.9 - 1.0	1192	0	159	1351	88.2	0.0

Table 5.6: Retrieval in the system of the Hopfield network with competitive feedback

Initial dist.	Success	Failure	Inconcl.	Total	Success(%)	Fail(%)
0.1 - 0.2	4	4	11	19	21.1	21.1
0.2 - 0.3	344	16	76	436	79.0	03.7
0.3 - 0.4	1062	0	155	1217	87.3	00.0
0.4 - 0.5	1121	0	154	1275	87.9	00.0
0.5 - 0.6	1201	0	170	1371	87.6	00.0
0.6 - 0.7	1190	0	263	1353	88.0	00.0
0.7 - 0.8	1260	0	156	1416	89.0	00.0
0.8 - 0.9	1379	0	186	1565	88.1	00.0
0.9 - 1.0	1178	0	170	1348	87.4	00.0

patterns are recognized correctly but none are recognized incorrectly. However, preliminary results we have obtained suggest that to obtain a buffer of useful width, too many correct recognitions at high initial overlap have to be sacrificed.

Chapter 6

Pattern Recognition by Searching the Periodic Orbits

The fact that chaotic systems have a huge collection of periodic points, each of which may be used as stored patterns, makes chaotic neural networks worth studying. However, one has to be careful. It is possible that the dense periodic points of a chaotic system are clustered in a small region creating practical difficulties to use them. The riddled basin boundaries may also cause problems. We focus on the class of systems in which there is a large number of well distributed periodic orbits, specially those of short periods. Pattern recognition in such chaotic systems can be realized if we augment the dynamics by a scheme that seeks the periodic points and vanishes after the periodic points are found. The procedure (see below), is meant to find the periodic points but not to alter their number, location or nature.

6.1 The Model Dynamical System

For the purpose of demonstration, we consider the "maximum hyperchaotic system" which has recently been studied by ref. [43]. This model consists of N neurons whose states are described by $S_i(t), i = 1, \ldots, N$. The N neurons are connected to one another through the synapses J_{ij}, $i, j = 1, \ldots, N$. The dynamics of the states is described by

$$S_i(t+1) = f(\sum_{j=1}^{N} J_{ij} S_j(t)), \quad i = 1, \ldots, N, \tag{6.1}$$

with

$$f(z) = \tanh(\alpha z)e^{-\beta z^2}, \tag{6.2}$$

where α and β are constants. Depending on J_{ij} and the parameters α and β, the system presents chaotic or nonchaotic behaviors. For example, when $N = 4$, $\alpha = 3$ and $\beta = 2$, the system is hyperchaotic which can be verified by checking the Lyapunov exponents in the phase space. For the synapses

$$J = \begin{pmatrix} 0.01 & 1.50 & 0.03 & 0.01 \\ 1.60 & 0.01 & 0.10 & 1.00 \\ 0.03 & 0.02 & 0.01 & 1.50 \\ 0.00 & 1.00 & 1.70 & 0.01 \end{pmatrix}, \tag{6.3}$$

all the four Lyapunov exponents of this 4-dimensional system are positive [43]. The neural network is thus "maximum hyperchaotic".

This hyperchaotic system with $N = 4$, $\alpha = 3$, $\beta = 2$ has infinitely many unstable periodic points embedded in a finite phase space. However, only the ones with short periods are better suited for pattern recognition. For pattern recognition, those chaotic systems that support a large number of well distributed periodic orbits of short periods should be preferable. The model system of this work is not claimed to be ideal for this purpose. We have chosen this model to show that the process of pattern recognition by a chaotic system is not affected by its number of positive Lyapunov exponents. Since there are no attracting periodic points, it is unlikely that a randomly chosen initial state variables will exactly match those of a periodic point for the dynamics to end in a persistent periodic behavior. Thus, for pattern recognition, we need to create "virtual basins of attraction" of the periodic points by augmenting the dynamics in a way that does not change the number, location or nature of the periodic points. This is easily done by an algorithm that finds the periodic point that lies in the nearest vicinity of a given initial point. All the states in this virtual basin of attraction will be identified as the pattern represented by this periodic point. If a periodic point cannot be reached from an initial state, then that state will be categorized as "I do not know". There are algorithms in the literature for finding periodic points of dynamical system. For the purpose of illustration, we use here the classic Newton's root finding method (NRFM) [44]. Using NRFM with $N = 4$, $\alpha = 3$, $\beta = 2$, we have found 25 fixed points of Eq. (1) (roots

of $S_i - f(\sum_{j=1}^{N} J_{ij} S_j(t)) = 0, \quad i = 1, \ldots, 4)$. These fixed points are denoted by the vectors $\mathbf{Y}_i = (S_1^{F[i]}, S_2^{F[i]}, S_3^{F[i]}, S_4^{F[i]})$ where $S_j^{F[i]}$, $j = 1, 2, 3, 4$ are the four fixed-point values of the state variables for the i^{th} fixed point.

Table 6.1: The fixed points

i	$S_1^{F[i]}$	$S_2^{F[i]}$	$S_3^{F[i]}$	$S_4^{F[i]}$
1	-0.5186	-0.1289	-0.5190	-0.1282
2	-0.1800	-0.6076	-0.3302	-0.6468E-1
3	-0.2428	-0.5491	-0.3536	-0.7057E-1
4	-0.5955	-0.1952	0.1227	0.4119E-1
5	0.2386	0.5679	-0.3522	-0.9470E-1
6	0.1551	0.4656E-1	-0.5736	-0.1758
7	-0.32983	-0.6865E-1	-0.1763	-0.6110
8	0.0	0.0	0.0	0.0
9	0.4224	0.4342	-0.3641	-0.4786
10	0.4005	-0.1048	0.2215	0.5846
11	-0.3158	-0.8375E-1	0.4144	0.4367
12	-0.4756	-0.3918	0.2437	0.6919E-1
13	-0.3575	-0.7517E-1	-0.2542	-0.5388.

Table 6.1 contains 13 of these fixed points correct to four significant figures. All the fixed points were generated by a double precision routine. Twelve other fixed points are found by changing the signs of these twelve fixed points (except $\mathbf{Y}_8$). These fixed points are used below for pattern recognition. We have also found a large number of periodic points of higher periods but they are not discussed in this work.

6.2 Pattern Recognition

The phase space, in the presence of NRFM, is visualized as disjoint regions of fixed points with virtual basins of attraction. When the dynamics is run in this space with an arbitrary initial state, one of the three things happens: (a) The state evolves into a known fixed point and the pattern is recognized. (b) No fixed point is reached. This is the "I do not know" behavior of the

network. (c) A new fixed point different from the known ones is found. This new fixed point is added to the memory. In this way all the fixed points that are accessible by the algorithm can be found if the initial list of fixed points is not complete. One may say that the phase space has been partitioned into regions of dynamic persistence (fixed points with virtual basins).

We now discuss the pattern recognition by considering the fixed points as the patterns themselves. Our task is to retrieve the fixed points. The initial state, e.g. near $\mathbf{Y}_2$ (denoted by $\mathbf{y}_2 = (y_1^2, y_2^2, y_3^2, y_4^2)$), is defined by

$$\mathbf{y}_2 = \mathbf{Y}_2 + \mathbf{e}(R)r \tag{6.4}$$

where, $\mathbf{e}(R)$ is an unit vector whose direction varies according to a random number R and r is a real number. The second term in (6.4) thus gives the distortion of the fixed point $\mathbf{Y}_2$ in the form of a 4-dimensional hypersphere with radius r. Fig. 6.1 illustrates statistically the retrieval rates for the fixed points at various radii. The numbers on the concentric circles are the ratio (percent) of number of successful recognitions to the total probes for recognizing the fixed points. First, the retrievability depends on the noise (radius r) determining how far $\mathbf{y}_2$ is from $\mathbf{Y}_2$. Then, it depends on the patterns. For example, the retrievability for $r = 0.2$ is as high as 0.99% in Fig.6.1(a), while it is only 81% in Fig.6.1(c). This is a reflection of the location of the fixed points and their virtual basins generated by the NRFM. It may be reminded that the synaptic connection matrix of our model is asymmetric. We would like to stress that it is the root finding method (in this paper NRFM) that determines the fixed points and retrieves a fixed point from a state which has only partial information on it. Without using this approach, the fixed points cannot be found, although we know that they exist.

The scheme presented above for pattern recognition in chaotic systems is different from that used in nonchaotic models. In the usual nonchaotic models patterns are stored in synapses (e.g. [6, 8, 11]) such that the dynamics alone drives an initial state having partial knowledge of a pattern to a fixed point (pattern). For chaotic networks, we need an additional algorithm such as the NRFM to drive a state to a fixed point (pattern). Thus, with the help of schemes such as the NRFM, chaotic systems can be used for pattern recognition. Compared to nonchaotic systems, the role of synapses

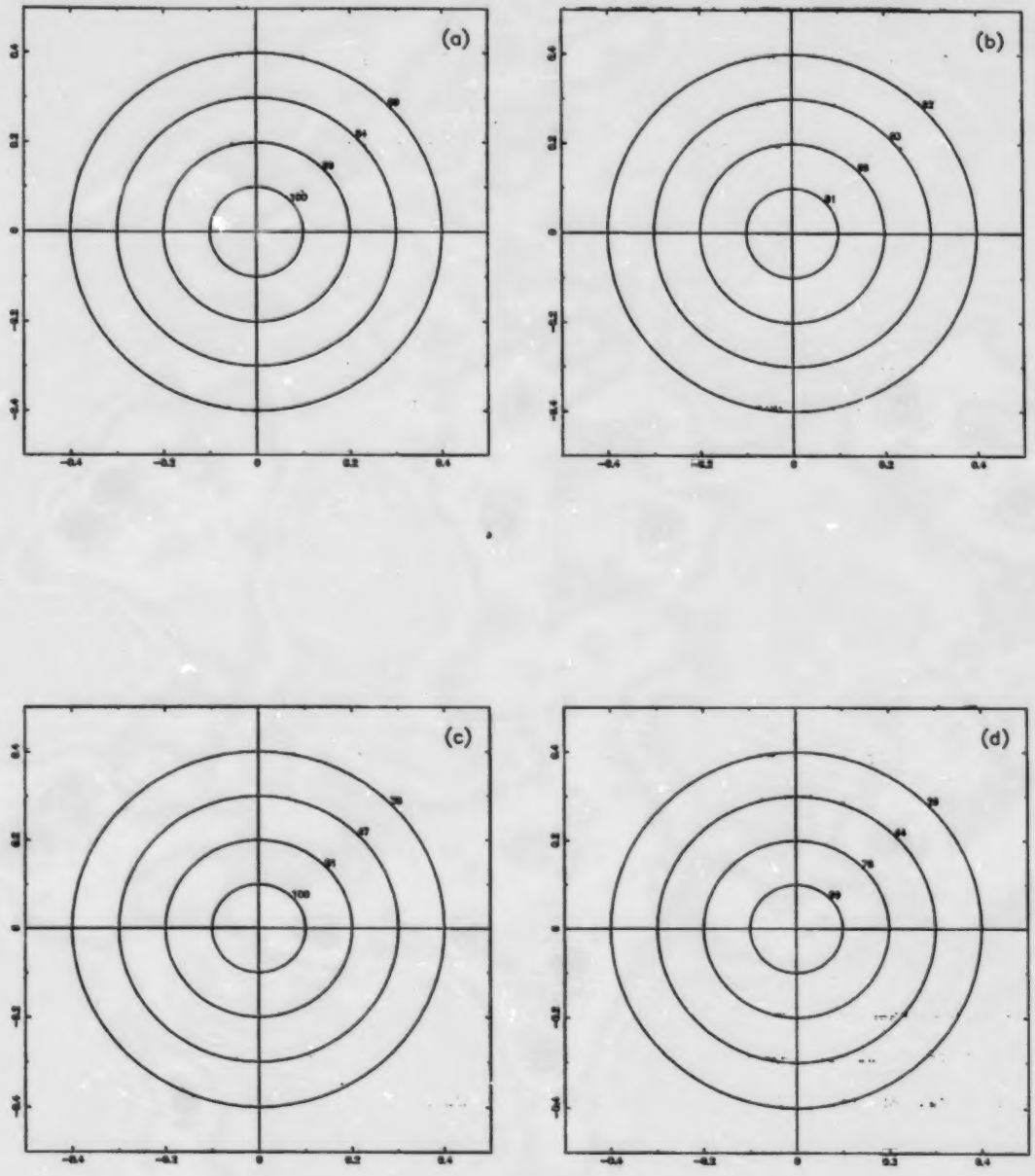

Figure 6.1: The retrieval of the fixed point (a) $\mathbf{Y}_1$; (b) $\mathbf{Y}_2$; (c) $\mathbf{Y}_5$; (d) $\mathbf{Y}_{10}$. The radii of the circles from inner to outer are $0.1, 0.2, 0.3$ and 0.4.

on patterns is less direct. At this stage, we can only say that the synapses (along with other parameters) influence the number and locations of the distinct fixed points which we have considered as patterns. Search for fixed points (periodic orbits) in dynamical systems is treated in this work as a process of pattern recognition. One advantage of using chaotic systems for pattern recognition is that the storage capacity can be increased enormously. With suitable higher dimensional chaotic systems pattern recognition using chaotic systems may be very useful for practical purposes. In this paper, we have categorized the fixed points of our model system as patterns. The question of mapping a given geometric shape into the fixed points remains to be addressed.

Chapter 7

Conclusions

We have been working on ways to exploit chaos for pattern recognition. Although our task is challenging, it has the potential of a big reward. The report contains an account of the progress made in the work so far.

In this report, we have reviewed (Chapters 2-5.1) the associative memory in some important chaotic neural networks and have presented (Chapters 5.2-6) our own work. We have discussed the models and the techniques used in them for associative memory.

Based on our work, it becomes clear that chaos needs to be controlled one way or another for retrieval of specific memories when partial information about these memories is available. Without control, a trajectory of a chaotic network will wander in the phase space visiting the sites of periodic points (which may be associated with patterns). Such visits to the neighborhoods of fixed points or periodic orbits may be characterized as "scanning the patterns" but no particular pattern will be recognized without appropriate control.

As for our own research, we have presented two distinct methods for pattern recognition. It is important to point out that both of our methods are applicable to chaotic and nonchaotic networks. In a way, it may be said that we have been able to generalize the concepts and techniques of pattern recognition.

In one of our approaches to control of chaos, we have used the technique of synchronization of chaotic trajectories. In this approach, we do not need

the existence of fixed points for pattern recognition. Here, the history of a trajectory that contains the information of a pattern is used to recognize an unknown pattern. This method has an added advantage in cases where short-term memories are important. The method works irrespective of the presence or absence of periodic orbits.

Since a chaotic network has a large number of unstable fixed points each one of which can be associated with a pattern, we have focussed on utilizing these fixed points for pattern recognition. To achieve our goal, we had to augment the dynamics by an algorithm that seeks the fixed points without altering their number, location or nature. By using the periodic orbits of chaotic networks, one can store enormously large number of patterns. It may be pointed out that the usual nonchaotic networks are plagued by low capacities. This technique works for both chaotic and nonchaotic systems.

MORE WORK NEEDS TO BE DONE.

Bibliography

[1] C. A. Skarda and W. J. Freeman, Behavior and Brain Science **10**, 161 (1987).

[2] W. J. Freeman, Biol. Cybernetics, **56**, 139 (1987).

[3] J. Glanz, Science, Vol. **277**, 1758 (1997).

[4] J. A. Anderson, *An introduction to Neural Networks*, Mit Press (1995).

[5] M. C. Mozar, M. I. Jordan and T. Petsche eds., *Advances in Neural Information Processing System 9* Mit Press (1997)

[6] J. J. Hopfield, Proc. Nat. Acad. Sci. USA Vol. **79**, 2554 (1982).

[7] D. J. Amit, H. Gutfreund and H. Sompolinsky, Phys. Rev. A**32**, 1007 (1985).

[8] E. Gardner. J. Phys. A**21**, 157 (1988)

[9] T. B. Kepler and L. F. Abbott, J. Physique **49**, 1657 (1988).

[10] Z. Tan and L. Schülke, Inter. J. Mod. Phys. B**26**, 3549 (1996).

[11] Z. Tan and M. K. Ali, Phys. Rev. (in press, 1998)

[12] K. Aihara, T. Takabe and M. Toyoda, Phys. Lett. A**114**, 333 (1990).

[13] M. Adachi and K. Aihara, 1997 International Symposium on Nonlinear Theory and its Application (Nolta'97), 645; M. Adachi and K. Aihara, Proc. ICONIP'97, 190.

[14] M. Adachi and K. Aihara, Neural Network **10**, 83-98 (1997).

[15] T. Ikeguchi, M. Adachi and K. Aihara, ????

[16] T. Nagashima, J. Miyazaki, Y. Shiroki and I. Tokuda, International Journal of Chaos Theory and Application **2**, 1-24 (1997).

[17] T. Nagashima and S. Matsuuchi, Nolta'97, 669 (1997);

[18] I. Tsuda, Neural Network **5**, 313 (1992).

[19] V. V. Kapelko and A. D. Linkevich, Phys. Rew. E**54**, 2802 (1996).

[20] S. Nara, P. Davis, M. Kawachi and H. Totsuji, International Journal of Bifurcation and Chaos **5**, 1205 (1995).

[21] S. Nara and P. Davis, Progress of Theoretical Physics, **88**, 845 (1992).

[22] M. Kushibe, Y. Liu and J. Ohtsubo, Phys. Rev. E**53**, 4502 (1995).

[23] Z. Tan and M. K. Ali, paper submitted

[24] M. K. Ali, Z. Tan, B. Hepburn and C. Tucker, paper submitted.

[25] M. Hasagawa, T. Ikeguchi and K. Aihara, Phys. Rev. Lett. **79**, 2344 (1997).

[26] J. K. Ryeu and H. S. Chung, Neurocomputing **13**, 281 (1996).

[27] J. J. Hopfield and D. W. Tank, Biol. Cybernet. **52**, 144 (1985).

[28] S-I. Amari. IEEE transactions on computers, Vol. c-**21**, No. 11, 1197 (1972).

[29] C. Grebogi, E. Ott, J. A. Yorke, Phys. Rev. Lett. **48**, 1507 (1992).

[30] E. Ott, C. Grebogi and J. A. Yorke, Phys. Rev. Lett. **64**, 1196 (1990)

[31] T. Shinbrot, E. Ott, C. Grebogi and J. A. Yorke, Phys. Rev. Lett. **65**, 3215 (1990).

[32] K. Pyragas, Phy. Lett. A**170**, 421 (1992).

[33] M. K. Ali and J-Q Fang, Phys. Rev. E**55**, 5285 (1997)

[34] J. W. Clark, Phys. Rep. **158**, 91 (1988).

[35] M. Bouten, A. Engel, A. Komoda and R. Serneels, J. Phys. **A23**, 4643 (1990).

[36] F. W. Milonni, M-L. Shih and J. R. Ackerhalt, *Chaos in Laser-Matter Interaction*, World Scientific Publishing Co Pte Ltd. (1987).

[37] A. J. Lichtenberg and M. A. Lieberman, *Regular and Stochastic Motion*, Springer-Verlag (1983).

[38] S. A. Barton, Automatica **31**, 1149 (1995)

[39] S. A. Barton, DRES SR-589 (1993)

[40] Private communications with S. A. Barton.

[41] Z. Tan and M. K. Ali, Progress Report - Dec 1997.

[42] Z. Tan and M. K. Ali, Progress Report - Sept 1997.

[43] J. W. Shuai, Z. X. Chen, R. T. Liu and B. X. Wu, Phys. Rev. **E56**, 890 (1997).

[44] W. H. Press, B. P. Flannery, S. A. Teukolsky and W. A. Vetterling, chapter 9, *Numerical Recipes*, Cambridge University Press (1989).

Report 4

The Associative Memory Based on Synchronization

(April - June, 1998)

Contents

1 Introduction **3**

2 The Model with Refractory (Delayed) Mechanism **5**
2.1 The Refractory Process 5
2.2 The Chaotic Recurrent Model 6

3 The Associative Memory **10**
3.1 Dependence on the Initial Condition 12
3.2 Dependence on Storage Capacity 15

4 Phase-synchronization in Chaotic Lattices **18**
4.1 Pattern Lattices and Phase Synchronization 18
4.2 Results 20

5 Conclusions **22**

Chapter 1

Introduction

Up to now, we have discussed the associative memory in a few neural networks with chaotic dynamics [1, 2, 3]. The synchronization has been studied in the Gardner model [4], Simon's model [5, 6, 7], the Hopfield model [3], and other chaotic systems [3]. These studies show that synchronization could be developed as an effective approach to the process of pattern recognition in chaotic neural networks, as chaos may be unavoidable or it may even be necessary in real biological systems [8, 9, 10]. Although there are a large number of studies about chaos in neural networks, it is still lacking general and effective methods to show how chaotic systems recognize patterns. Pattern recognition is an important part of information processing in any biological system. In our preceeding work, we have tried to apply synchronization to recognition and have shown that this approach works in the Gardner model, one well-known spin-glass-like model. In order to develop this method, we need to test and improve it in other models. Our goal is to find a general approach to pattern recognition in chaotic neural networks.

In this report, we turn to a chaotic recurrent model developed by Aihara *et al* and lattices of chaotic elements. The former model has a biological basis [11, 12] and has been discussed widely [13, 14, 16, 17, 15]. However, the pattern recognition capability of this model has not been really discussed. What has been discussed is the "searching process" in the phase space. This process refer's to the random visitation of stored patterns in the state space starting from an initial state. This searching process neither tells us any information about what pattern will be recognized nor its connection with

the initial state. Therefore, it is not enough to address the associative memory of this model without discussing the mechanism of pattern recognition. Pattern recognition means that when a pattern is recognized by the system, the dynamics should not drive the system away from the state correponding to the pattern, or at least from the viewpoint of statistics, the system should return to that state with overwhelming probability.

Any chaotic system will not have this property. Starting form any initial state, the system evolves throughout the whole phase space after a long enough time. The dynamics is divergent in other words, and will not drive the system to a fixed state. This is the essential difference between the chaotic and nonchaotic systems. How can we construct a mechanism to counteract this? We need an external force. This will be realized by imposing synchcronization.

Another feature that should not be ignored is the initial condition. We believe that even in the chaotic system, the dynamics of the early steps still include information on the initial condition. With the evolution, the information disappears. Of course, the loss of information in a chaotic system is much faster than in a nonchaotic system. Therefore, we have to concentrate on the early dynamics if we want to utillize the information hidden in the initial state. We refer to this as short-term memory.

In this report we first use trajectory control in a chaotic model with refractoriness. Then, we study the phase synchronization in chaotic lattices. We found that the synchcronization approach is effective for pattern recognition. The details will be given in the following.

Chapter 2

The Model with Refractory (Delayed) Mechanism

One widely discussed model is the chaotic recurrent model proposed by Aihara *et al* [11]. In this model, feedback and refractory processes are taken into account. Therefore, we first discuss briefly the background of the refractory process in neurons of real biological systems. The association aspect will be discussed after the introduction.

2.1 The Refractory Process

The refractory process describes the effect of the motion of ions in *past* time on the motion of the ions at the *current* time. There is biological evidence for this phenomena [12]. A real neuron has a membrane that possesses negative voltage. Inside and outside the neuron there are different concentrations of sodium and potassium ions. The flow of ions is determined by their concentration gradients and the electrical potential. Equilibrium of flow of the potassium ions can be reached by these two processes. However, equilibrium of the sodium ions can be broken, or depolarized, by adding a positive external stimulus (voltage or current) to the neuron. Such depolarization changes the concentration of ions inside and outside the neuron. When the stimulus is above a threshold, the neuron generates a spike action potential (firing). After each firing, a neuron will go back to the original equilibrium, until the next firing. The frequency of the firing, not the strength of the firing, is a

function of the external stimulus. The concept of refractoriness means that for a brief period it is *not* possible to evoke a second action potential. There is a quiet period (relative refractory period), during which the neuron will not fire. The quiet period is a consequence of the earlier firing (the motion of ions), i.e. the past motion of the ions influences the actual and post movement.

People believe that refractoriness is an important principle for the modelling of artificial neural networks. We will see in the next section one proposal for taking this mechanism into account.

2.2 The Chaotic Recurrent Model

A naive model of a ***single*** refractory neuron assumes that the refractory process decreases exponentially with time and so the dynamics of the neuron can be written

$$s(t+1) = f\left[A(t) - \alpha \sum_{r=0}^{t} k^r s(t-r) - \theta\right], \tag{2.1}$$

where $s(t)$ is the state of the neuron at time t, $A(t)$ is the input, and the refractory factor, k^r with $0 < k < 1$, decreases exponentially with time. f is the action potential and can be given in different forms.

Eq. 2.1 can be reduced to

$$s(t+1) = f(y(t+1)) \tag{2.2}$$

with

$$y(t+1) = ky(t) - \alpha f(y(t)) + a(t) \tag{2.3}$$

and

$$a(t) = A(t) - kA(t-1) - (1-k)\theta. \tag{2.4}$$

The neuron has chaotic behaviour, as can be verified by checking the Lyapunov exponent. Fig. 2.1 gives the exponent as a function of α, using the parameters and potential given in the caption. Over some range of α the exponent is positive, implying the existence of chaos. From the complicated form of the figure one should expect rich dynamic behaviour. Such single

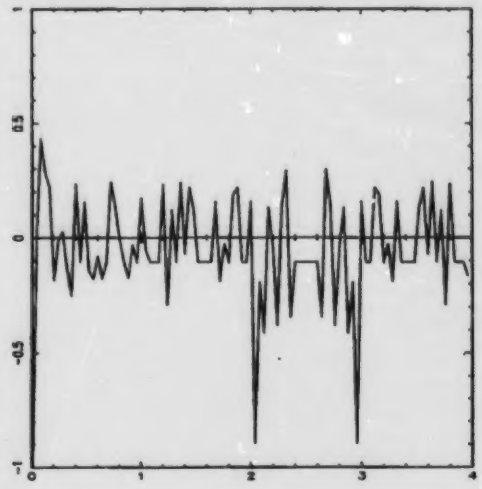

Figure 2.1: The Lyapunov exponent of the dynamics of single neuron with refractory process. $y(0) = 0.5, k = 0.9, \alpha = 5.0$ and $f(y) = (1 + exp(-y/0.015))^{-1}$.

neurons can be used to compose networks according to the following equations [11]

$$s_i(t+1) = f(\eta_i(t+1) + \zeta_i(t+1)), \tag{2.5}$$

$$\phi_i(t+1) = k_e\phi(t) + \sum_{j=1}^{M} v_{ij}A_j(t), \tag{2.6}$$

$$\eta_i(t+1) = k_f\eta_i(t) + \sum_{j=1}^{N} w_{ij}s_j(t), \tag{2.7}$$

$$\zeta_i(t+1) = k_r\zeta_i(t) - \alpha s_i(t) + a_i, \tag{2.8}$$

$$f(z) = [1 + \exp(-\frac{z}{\epsilon})]^{-1}, \tag{2.9}$$

where ϵ is a positive constant determining the steepness of the potential function. The ideas behind (2.5 – 2.9) must be explained. $s_i(t)$ is the output of neuron i at time t. $\phi_i(t)$ is the external input to the neuron i at time t with v_{ij} the connection weight from the jth external input to the ith neuron.

There are M neurons involved in the communication with the outside. $\eta_i(t)$ is the input from all other neurons connected by the synapses w_{ij} (defined below) with a feedback process of strength k_f. $\zeta_i(t)$ represents the refractory mechanism of neuron i at time t and a feedback of strength k_r and α is also carried out in this process: the neuron will generate an action potential through $f(z)$, when the accumulated feedback terms in $\zeta_i(t)$ exceed the threshold a_i. The threshold a_i, the feedback refractory strength k_r and α guarantee that neuron i will not stay in one state for long as it makes the neuron change constantly. Even when the initial values of s_i, η_i and ζ_i are null, the nonzero threshold will yield nonstatic output from neuron i. Therefore, it is the introduction of ζ_i that generates chaotic behavior, while the η_i terms produce the usual characteristics of the nonchaotic model. The competition between these two forces results in rich dynamic behavior. This behavior depends of course on the parameters involved in eqs. (2.5-2.9). In order to simplify the discussion, we ignore the contribution from $\phi_i(t)$. This amounts to assuming that the external input is constant. The contribution will be restored later, when we discuss pattern recognition.

The synapses connecting neurons are defined by the Hebbian rule

$$w_{ij} = \frac{1}{2P} \sum_{\mu=1}^{P} (\xi_i^\mu - \overline{\xi})(\xi_j^\mu - \overline{\xi}), \tag{2.10}$$

where, $\xi_i^\mu, \mu = 1, \ldots, P$, are the patterns stored in this network. They are assigned randomly to be 0 or 1. $\overline{\xi}$ is set to be 0.5.

Fig. 2.2 shows such behaviors for parameters, by giving initial $s_i(t), \eta_i$ and ζ_i zero. For the parameters used in the figure, Fig. 2.2(a) represents chaotic behavior, while Fig. 2.2(b) shows a stable output (that is 1). Therefore, the neuron has chaotic output in Fig. 2.2(a), but not in Fig. 2.2(b). Actually, the property is proven by Ikeguchi *et al* [14], after calculating the Lyapunov exponents in this system. They found that the largest exponents is positive for a network with 16 neurons with $N = 16, p = 4$ and the parameter used in Fig.2.2(a).

The effect of the threshold a_i and feedback parameter α is also related to the dynamic property of the network. One can find the details in the literature. In short, the dynamic behavior depends on the parameters. A typical

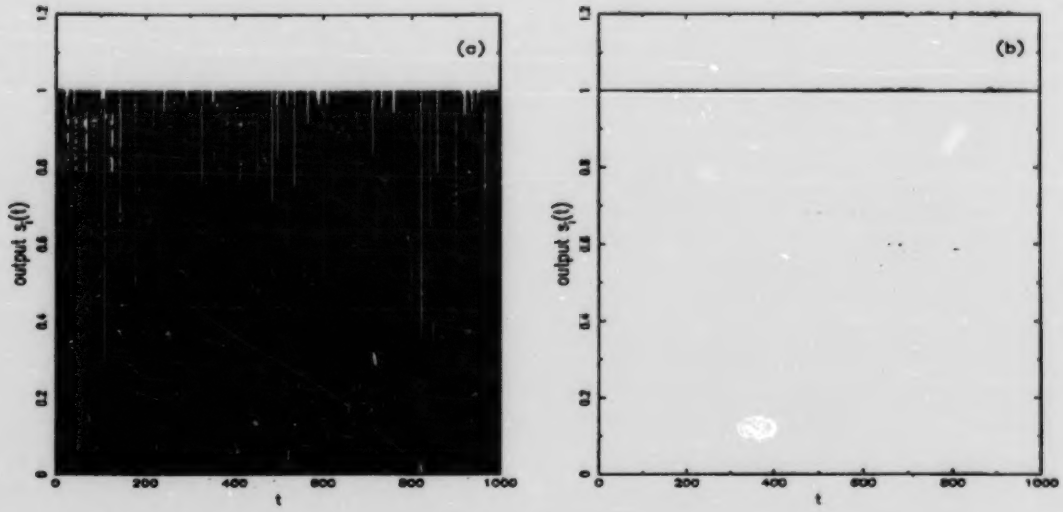

Figure 2.2: The output $s_i(t)$ vs. t in the network with $N = 16, p = 4, k_f = 0.2, a_i = 1., \epsilon = 0.015$ and (a) $k_r = 0.9, \alpha = 5.0$; (b) $k_r = 0.1$, $\alpha = 0.5$. The dynamics in (a) ((b)) is chaotic (nonchaotic).

set of the value of the parameters is $k_f = 0.2, k_r = 0.9, \alpha = 5.0, a_i = 1.0$ and $\epsilon = 0.015$. There are of course regions where the dynamic behavior is a mixture of chaotic and nonchaotic output.

Chapter 3

The Associative Memory

Adachi, Ikeguchi and Nagashima's groups used the above model to explore pattern recognition [13] – [15]. Let us store 4 patterns in the net. The synapses are determined by Eq. (2.10) with $P = 4$. The main result is that the network first presents nonperiodic behavior in a transient phase, after which, in most cases, the network has periodic behavior. The nonperiodic behavior in the transient phase is described as "random wandering" in the phase space. However, there is no evidence to show that any pattern can be recognized, although the stored patterns may also be visited during the random movement. For example, if we check the retrieval rate of a pattern, the rate is low. More seriously, even the probability of visiting a pattern, however closely the dynamics start, does not have a relatively large value. This behavior does not give any information on the pattern recognition. Therefore, it is necessary to introduce new approaches to investigate the ability of recognition.

As discussed in Chapter 1, the information about initial conditions in a chaotic system may be retained for the first few dynamic steps. As the system evolves, this information is lost. Our strategy is thus to extract the infornmation before it is lost. To this end, we turn to chaos control.

It is well known that chaotic trajectories can be synchronized (stabilized) to desired trajectories. Synchronization of trajectories may provide an effective method which is completely different from the usual concepts in pattern recognition. The synchronization of chaotic trajectories itself is not unique

[30] – [34]. Here we utilize the feedback method proposed in [33] and select the trajectory stimulated initially by a pattern (e.g. ξ^1) as the desired orbit and synchronize other chaotic trajectories to this orbit. We try to get useful information during the synchronization.

We denote the desired trajectory as $\zeta^1(t)$ with $\zeta^1(0) = \xi^1$ and a trajectory to be synchronized as $s(t)$ which is near ξ^1 at $t = 0$. The relation between $\zeta^1(t)$ and $s(t)$ is measured by a dynamic distance

$$d^1(t) = \left\langle \frac{1}{N} \sum_{i=1}^{N} (s_i(t) - \zeta_i^1(t))^2 \right\rangle_\xi , \tag{3.1}$$

where, $\langle \cdots \rangle_\xi$ means the average over random patterns ξ in the Gardner model [4, 21]. $d^1(0)$ gives the initial distance. Although $\zeta^1(t)$ is not ξ^1, it is generated by ξ^1 and in a deterministic system it is unique and repeatable (even though the trajectories depend on the machine and numerical accuracy) if started again from ξ^1. Therefore, we "transfer" ξ^1 to $\zeta^1(t)$. $s(0)$ is recognized to be ξ^1 if $d^1(t)$ approaches zero.

In the chaotic model described in chapter 2, the dynamics will drive $s(t)$ away from $\zeta^1(t)$. $d^1(t)(t > 0)$ is larger than $d^1(0)$. But if at every dynamic step we "help" $s(t)$ to go a little bit closer to $\zeta^1(t)$, we may get useful results. This is the idea of chaos control through feedback. We now try to use this idea to perform recognition in a chaotic system.

The feedback is imposed by modifying the trajectory $s(t)$ at time t with

$$s_i(t) \to s_i^f(t) = s_i(t) + k(\zeta_i^1(t) - s_i(t)), \quad i = 1, \ldots, N \tag{3.2}$$

where $0 < k < 1$ is a constant called hereafter the feedback strength. For $k = 0$, there is no feedback; $k = 1$, the trajectory is modified immediately to be the desired one. The state is $s^f(t)$ after the feedback process is utilized for the next dynamics. Because the feedback is applied directly and at every time step on the output of the system, $s^f(t)$ will be driven to $\zeta^1(t)$ after a sufficiently long time, depending on k. After synchronization, the feedback term goes to zero and the distance in Eq. (3.1) vanishes (or becomes very small). However, we cannot be sure that $d^1(t) = 0$ in this case means that $s(t)$ is recognized, because $s(t)$ can be synchronized in the same way with all

other trajectories $\zeta^2, \ldots, \zeta^p$ coming initially from other patterns, $\xi^2, \ldots, \xi^p$, respectively. Therefore, we need another criterion to distinguish between the synchronized orbits.

The criterion that we have used in this work is the time of synchronization. We believe that the shortest time of synchronization tells us which pattern the state $s(0)$ should be identified with. More exactly, we denote $d^\mu(t)$ as the distance by replacing index 1 in Eq. (3.1) with μ. The shortest time

$$t^r = min\{t^1, \ldots, t^p\}, \tag{3.3}$$

where t^μ satisfy

$$d^\mu(t^\mu) < d_0, \quad \mu = 1, \ldots, p, \tag{3.4}$$

determines that pattern r is the target pattern and d_0 is a small number. If the shortest time t^r does not exist ($d^\mu(t^\mu)$ are same for all patterns before d_0 is reached), then $s(0)$ can not be recognized. In practice we do not have to compare all distances, because after the average in Eq. (3.1) all patterns are equivalent. Below we set $s(0)$ to be related to ξ^1 with the initial distance $d^1(0)$ and compare the distance between $s(t)$ and $\zeta^1(t)$ and that between $s(t)$ and an arbitrary trajectory, see $\zeta^2(t)$.

Based on the above disscussions, we studied the model and found that the ability of recognition is related to the initial distance, the storage capacity and the size of the network. In the following, we disscuss each relationship seperately.

3.1 Dependence on the Initial Condition

We set $N = 100$ in Eq. (2.7) and $P = 4$ in (2.10). The parameters are chosen to be $\alpha = 10, a_i = 2, \epsilon = 0.015$, $k_f = 0.2$ and $k_r = 0.9$. The model with this set of parameters was shown to be chaotic and the net wanders randomly in the phase space. Now we stimulate the system with patterns, see ξ^1 and ξ^2, and a state $s(0)$ near ξ^1 with the initial distance $d^1(0)$. Then during the evolution, we get three trajectories in the phase space, $s(t)$ from $s(0)$, $\zeta^1(t)$ from ξ^1 and $\zeta^2(t)$ from ξ^2. We compare the dynamic distance $d^1(t)$ between $s(t)$ and $\zeta^1(t)$ and $d^2(t)$ between $s(t)$ and $\zeta^2(t)$. Fig. 3.1(a), (b), (c) and (d) give the distances $d^1(t)$ and $d^2(t)$ versus t for initial distance $d^1(0) = 0.1$,

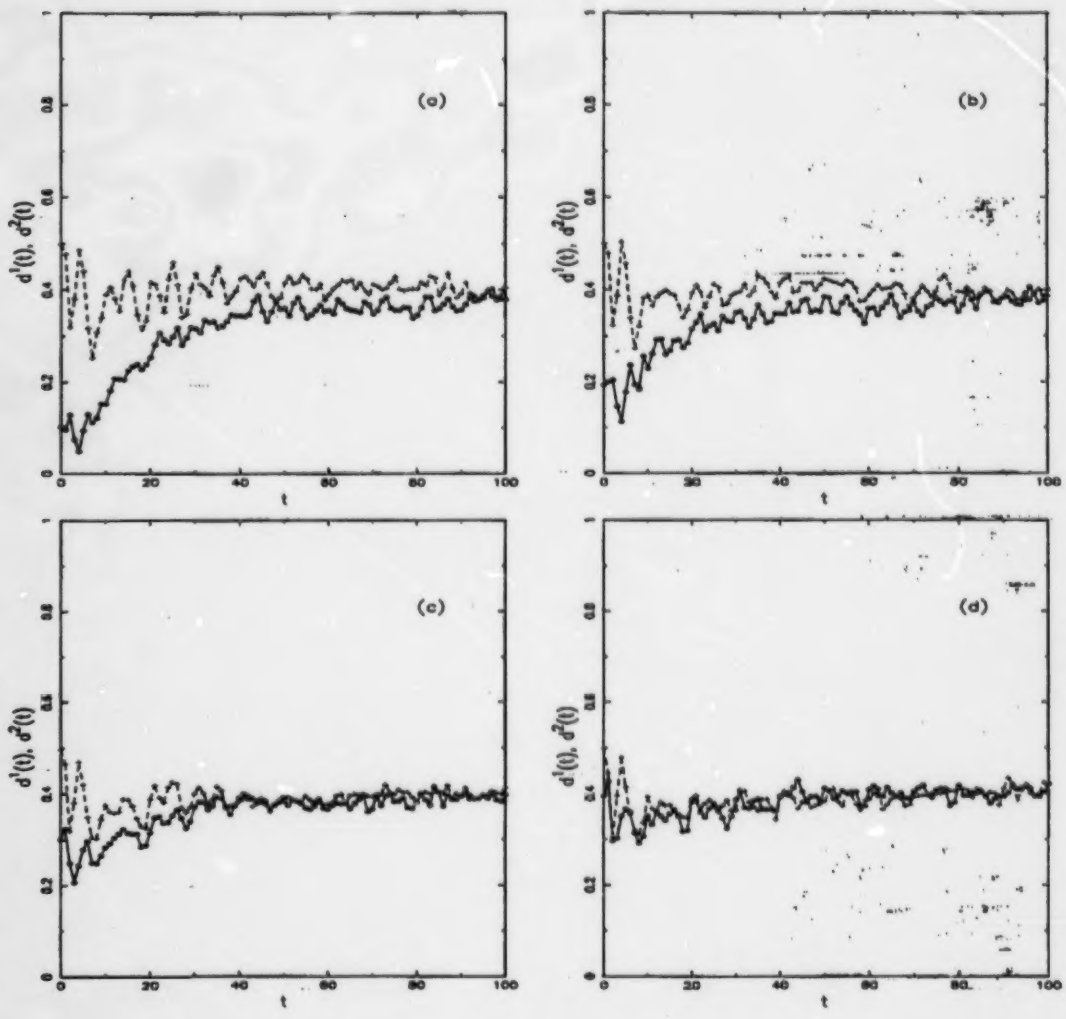

Figure 3.1: The dynamic distances $d^1(t)$ (circles) and $d^2(t)$ (crosses) vs. time t with $P = 4$, $N = 100$, $d_0 = 0.02$ and various $d^1(0)$: (a) $d^1(0) = 0.1$; (b) $d^1(0) = 0.2$; (c) $d^1(0) = 0.3$; (d) $d^1(0) = 0.4$. No synchronization is imposed. The points are linked for better insight.

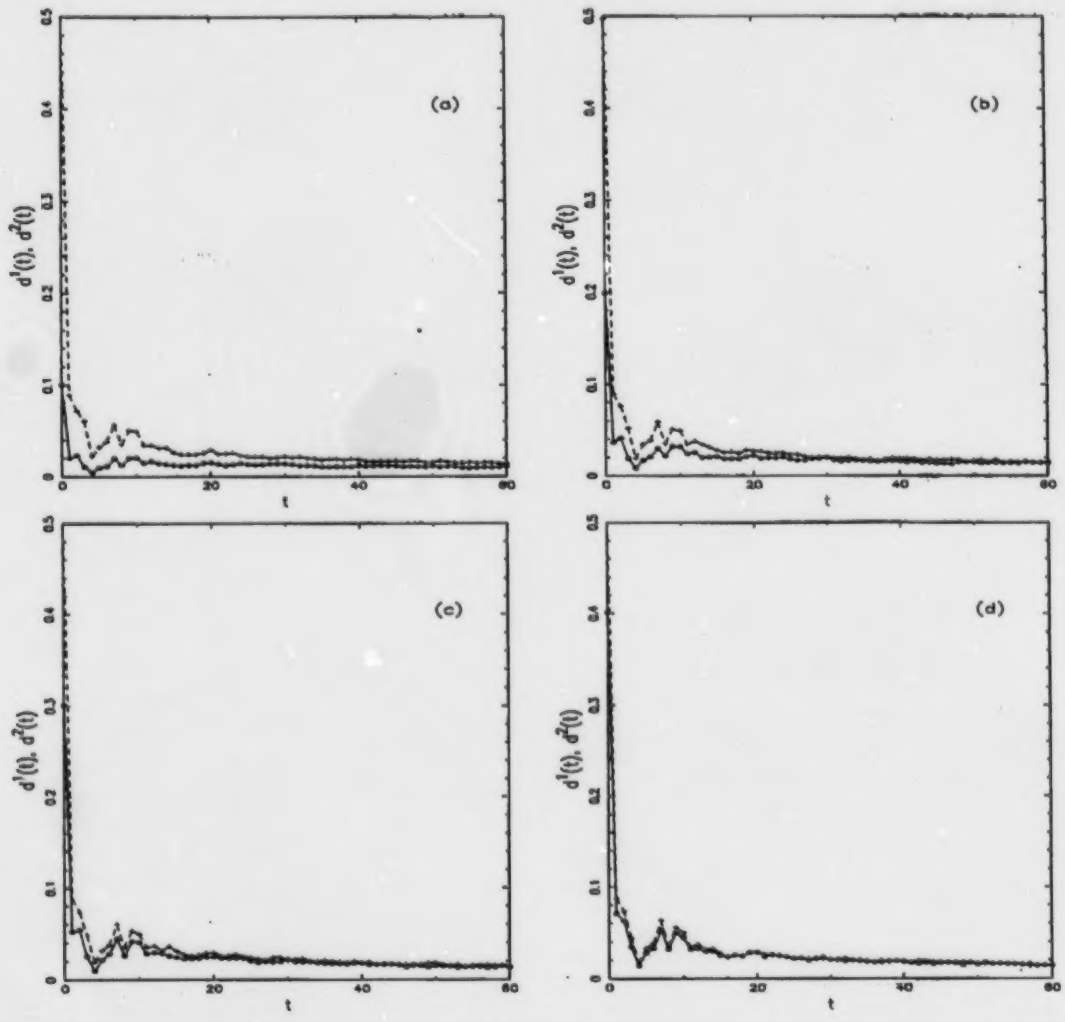

Figure 3.2: The dynamic distances $d^1(t)$ (circles) and $d^2(t)$ (crosses) vs. time t with $P = 4$, $N = 100$, $d_0 = 0.02$ and various $d^1(0)$: (a) $d^1(0) = 0.1$; (b) $d^1(0) = 0.2$; (c) $d^1(0) = 0.3$; (d) $d^1(0) = 0.4$. Synchronization with strength $k = 0.7$ is imposed. The points are linked for better insight.

0.2, 0.3 and 0.4, respectively. The strength of feedback k is set to be 0, while $P = 4$ and $N = 100$. The figures show that the distances go to the same value after a certain time, although in the short-term $d^1(t)$ is different from $d^2(t)$. This is an indication of the divergence of the chaotic system.

After synchronization is introduced, $d^1(t)$ and $d^2(t)$ become smaller and smaller and finally they tend to zero. Fig 3.2 shows the distances, after the feedback is imposed. When condition (3.4) is satisfied, the two distances $d^1(t)$ and $d^2(t)$ are distinguishable in Fig 3.2(a), (b) and (c), while in 3.2(d) they are indistinguishable. In Fig 3.2(a) and (b), the distances can be easily distinguished, while in (c) the difference is not as clear, but still recognizable. Therefore, according to the discussions in the preceding part in this chapter, pattern recognition is sucsessful in Fig 3.2(a), (b) and (c), while it has failed in Fig 3.2(d). Since the only difference among the figures is the initial distance $d^1(0)$, the above fact reflects the influence of the initial distance. When the initial distance is over a critical value the recognition will fail.

3.2 Dependence on Storage Capacity

The ablity of pattern recognition depends not only on the initial distance, but also on the storage capacity. Fig. 3.3 shows the effect of the capacity. We set $P = 20$. Compared with Fig. 3.2, Fig. 3.3 gives oscillating distances and small initial distances below which the pattern can be recognized. In Fig. 3.3(a) with $d^1(0) = 0.01$, $d^1(t)$ and $d^2(t)$ are clearly distinguishable when the condition (3.4) is valid, while in Fig. 3.3(b) and (c) with $d^1(0) = 0.03$ and 0.05 respectively, the difference is minor. In Fig. 3.3(d), there is no difference between the distances. Therefore, the recognition is sussesful in Fig. 3.3(a), (b) and (c) with very small initial distances, and it has failed in Fig. 3.3(d). Table 3.1 gives the relationship between P and critical $d^1(0)$ above which the recognition will fail. We see that when P is smaller, $d^1(0)$ can take a bigger value. We have investigated the system with different sizes and we have obtained the same qualitative results.

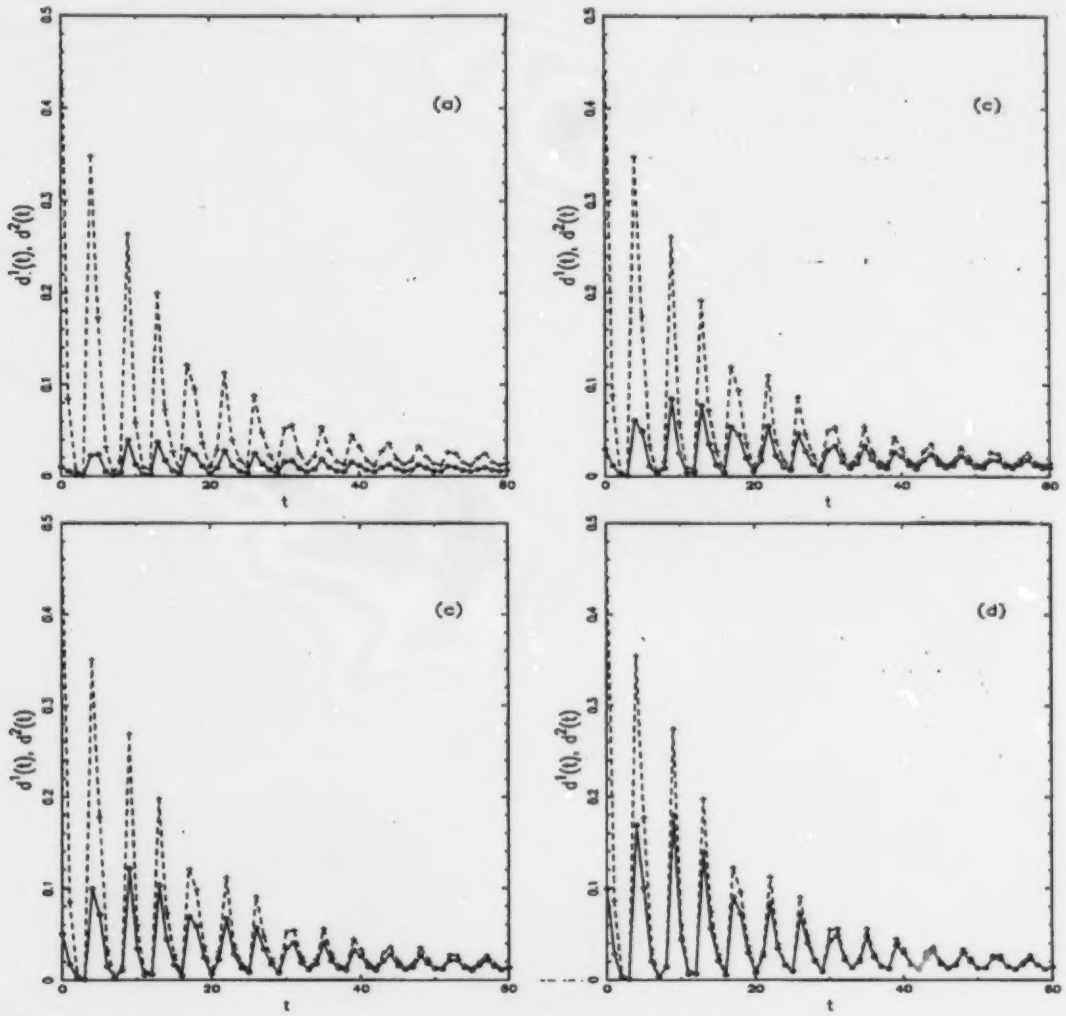

Figure 3.3: The dynamic distances $d^1(t)$ (circles) and $d^2(t)$ (crosses) vs. time t with $P = 20$, $N = 100$, $d_0 = 0.02$ and various $d^1(0)$: (a) $d^1(0) = 0.01$; (b) $d^1(0) = 0.03$; (c) $d^1(0) = 0.05$; (d) $d^1(0) = 0.1$. Synchronization with strength $k = 0.8$ is imposed. The points are linked for better insight.

P	4	10	20	30
$d^1(0)$	0.30	0.12	0.03	0.02

Table 3.1: The critical values of $d^1(0)$ versus P for $N = 100$.

Chapter 4

Phase-synchronization in Chaotic Lattices

We are investigating the use of coupled lattices of chaotic elements for pattern recognition. Each element in the lattice is a chaotic-oscillator, comprised of its own coupled sub-components. Our innovation is to use the Hebbian rule to determine the coupling strengths of a lattice according to one of the target patterns, which we call the parent pattern for that lattice, and then to designate a lattice for each of the patterns. The more similar two patterns are, the more similar their lattices are, and therefore their dynamics. We demonstrate one method of utillizing this dynamical similarity for pattern recognition.

4.1 Pattern Lattices and Phase Synchronization

The Hebbian rule dictates that, for a parent pattern vector $\xi_{1..N}$, the elements of the coupling matrix are

$$W_{ij} = \xi_i \xi_j, \quad i,j = 1,2,...N \tag{4.1}$$

where the ijth element is the coupling strength between the ith and jth lattice sight. For results presented here, only 1 sub-component of the chaotic

oscillator at each sight is coupled. The lattice elements are Lorentz oscillators. Therefore, with 1st component coupling, the dynamics are governed by the equations

$$\begin{aligned} \dot{y}_1^i &= 10(y_2^i - y_1^i) + \eta \sum_{j=1}^{N} W_{ij} y_1^j, \\ \dot{y}_2^i &= 28y_1^i - y_2^i - y_1^i y_3^i, \\ \dot{y}_3^i &= -\frac{8}{3} y_3^i + y_1^i y_2^i \\ & i = 1, 2, ...N \end{aligned} \tag{4.2}$$

where the number of lattice sights is N, and η is a coupling constant. The subscripts indicate the sub-component of the oscillator, while the superscripts indicate the lattice sight.

Using feedback between the different pattern lattices, we sought a characteristic of their synchronization that would serve as an indication of the similarity of their parent patterns. Two indicators we explored were the synchronization times when patterns were run together two at a time

$$Y_{slave} = Y_{slave} + \beta(Y_{master} - Y_{slave}); \tag{4.3}$$

and syncronization under distance-dependent competitive feedback to all patterns

$$Y_{slave} = Y_{slave} + \frac{1}{A} \sum_{\mu=1}^{p} \beta(\mu)(Y_\mu - Y_{slave}), \tag{4.4}$$

where p is the number of patterns and therefore the number of master lattices, each one indicated by the subscript μ. A is a normalization constant for the β's.

On the former method, the pattern with the shortest synchronization time is taken to be the recognized pattern. With the latter, all patterns are run in parallel, with the strength of the feedback coefficient for each master, $\beta(\mu)$, dependent on the Euclidian distance of that master's state vector to the state vector of the slave. The slave can synchronize with only one master, which is taken to be the recognized pattern.

However, *total synchronization* is too "coarse" for pattern recognition with non-identical chaotic systems, requiring very strong coupling if synchronization is to occur at all. With strong enough coupling, all lattices synchronize nearly simultaneously. With weak coupling, lattices of different parents will not synchronize, which is of course desired, but neither will a lattice with a damaged pattern for its parent synchronize with a lattice with the undamaged parent, regardless of how little damage is applied.

We have been able to demonstrate instead, the feasibility of pattern recognition through *phase synchronization*. Two non-identical lattices can be phase synchronized without the amplitude of their state vectors coinciding. For our Lorentz elements, we used the phase definition introduced by Pikovsky et al.[35]

$$\phi^i = arctan\left(\frac{u^i}{y_3^i}\right) \tag{4.5}$$

$$u^i = \sqrt{(y_1^i)^2 + (y_2^i)^2} \tag{4.6}$$

4.2 Results

Trials are conducted for 10 patterns with coupled lattices of 10 Lorentz elements, under direct feedback, equation(4.3). The dynamics were carried out with a fourth-order Runge-Kutta with the feedback added in as a correction after each iteration, and with a coefficient $\beta = .25$. A low coupling strength was used, ($\eta = .01$), which allows the Lorentz elements to remain in their chaotic domain. The pattern loading ($\equiv \frac{\text{number of patterns}}{\text{number of nodes}}$) can be considered to be either $\frac{1}{3}$, if you count each sub-element in the lattice as a node, or 1, if you count only the actual lattice sights.

The results of Table 4.1 show a clear dependence of synchronization times on pattern damage and high retrieval rates for low damages, demonstrating, in prinicple, the use of the times as indicators of pattern similarity. However, with non-chaotic nets, damages up to ± 0.5 can be achieved with greater than 50 percent success. With our chaotic nets, recognition drops off much more quickly with increasing damage, with almost no statistically significant

Damage +/-	Success Percentage	Damage +/-	Success Percentage
0.0000	100.0	0.0005	14.7
0.0001	37.2	0.0006	12.1
0.0002	25.8	0.0007	13.0
0.0003	18.9	0.0008	12.0
0.0004	17.1	0.0009	9.1

Table 4.1: The number of successful recoginitions with various pattern damages. The target pattern is damaged by adding the damage amount times a random number $\in (-1, 1)$.

recognitions for damages beyond 0.0009. Hence, our claim is only regarding the demonstration of the principle: similarity of behaviour is indicative, ***quantifiably***, of similarity of structure. It remains to be seen how best to use this principle for pattern recognition. Further investigations need to be made into the maximal coupling strength for pattern recognition, the effectivenesss of differential, as opposed to direct, feedback, and the effectiveness of competitive feedback.

Chapter 5

Conclusions

Chaos needs to be controlled for retrieval of specific memories when partial information about these memories is available. Without control, the trajectory of a chaotic network will wander in the phase space visiting the sites of periodic points. In this case, no particular pattern will be retrieved.

In this report, we treated two models with two distinct methods for pattern recognition. It is important to point out that both of our methods are applicable to chaotic and nonchaotic networks. In one of our approaches to the control of chaos, we have used the technique of synchronization of chaotic trajectories. In this approach, we do not need the existence of fixed points for pattern recognition. Here, the history of a trajectory contains the information of a pattern and is used to recognize an unknown pattern. This method has a particular advantage in cases where short-term memories are important. The method works irrespective of the presence or absence of periodic orbits.

In our second approach, we utilized phase sychronization among chaotic lattices. While many lattices running in parallel may not be (currently) feasible as a pattern recoginition paradigm, a demonstration of a *dynamic* criterion of pattern similarity is, we argue, necessary to furthering the field. In the present case, the similarity of the patterns, and hence of the lattices, results in a similarity of the dynamics. Chaotic lattices offer incredibly rich behaviour to work with, while phase synchronization allows us to measure the similarity of that behaviour dynamically.

Bibliography

[1] Z. Tan and M. K. Ali, Progress Report - Sept 1997.

[2] Z. Tan and M. K. Ali, Progress Report - Dec 1997.

[3] Z. Tan, B. Hepburn, C. Tucker and M. K. Ali, Progress Report - March 1998.

[4] Z. Tan and M. K. Ali, Phy. Rev. E (in press).

[5] S. A. Barton, Automatica **31**, 1149 (1995)

[6] S. A. Barton, DRES SR-589 (1993)

[7] Private communications with S. A. Barton.

[8] C. A. Skarda and W. J. Freeman, Behavior and Brain Science **10**, 161 (1987)

[9] W. J. Freeman, Biol. Cybernetics, **56**, 139 (1987).

[10] J. Glanz, Science, Vol. **277**, 1758 (1997).

[11] K. Aihara, T. Takabe and M. Toyoda, Phys. Lett. A**114**, 333 (1990).

[12] J. A. Anderson, *An introduction to Neural Networks*, Mit Press (1995).

[13] M. Adachi and K. Aihara, 1997 International Symposium on Nonlinear Theory and its Application (Nolta'97), 645; M. Adachi and K. Aihara, Proc. ICONIP'97, 190; M. Adachi and K. Aihara, Neural Network **10**, 83-98 (1997).

[14] T. Ikeguchi, M. Adachi and K. Aihara, ????

[15] T. Nagashima, J. Miyazaki, Y. Shiroki and I. Tokuda, International Journal of Chaos Theory and Application **2**, 1-24 (1997). T. Nagashima and S. Matsuuchi, Nolta'97, 669 (1997);

[16] M. Hasagawa, T. Ikeguchi and K. Aihara, Phys. Rev. Lett. **79**, 2344 (1997).

[17] J. K. Ryeu and H. S. Chung, Neurocomputing **13**, 281 (1996).

[18] M. C. Mozar, M. I. Jordan and T. Petsche eds., *Advances in Neural Information Processing System 9* Mit Press (1997)

[19] J. J. Hopfield, Proc. Nat. Acad. Sci. USA Vol. **79**, 2554 (1982).

[20] D. J. Amit, H. Gutfreund and H. Sompolinsky, Phys. Rev. **A32**, 1007 (1985).

[21] E. Gardner. J. Phys. **A21**, 157 (1988)

[22] T. B. Kepler and L. F. Abbott, J. Physique **49**, 1657 (1988).

[23] Z. Tan and L. Schülke, Inter. J. Mod. Phys. **B26**, 3549 (1996).

[24] I. Tsuda, Neural Network **5**, 313 (1992).

[25] V. V. Kapelko and A. D. Linkevich, Phys. Rew. **E54**, 2802 (1996).

[26] S. Nara, P. Davis, M. Kawachi and H. Totsuji, International Journal of Bifurcation and Chaos **5**, 1205 (1995); S. Nara and P. Davis, Progress of Theoretical Physics, **88**, 845 (1992).

[27] M. Kushibe, Y. Liu and J. Ohtsubo, Phys. Rev. **E53**, 4502 (1995).

[28] J. J. Hopfield and D. W. Tank, Biol. Cybernet. **52**, 144 (1985).

[29] S-I. Amari. IEEE transactions on computers, Vol. C-**21**, No. 11, 1197 (1972).

[30] C. Grebogi, E. Ott, J. A. Yorke, Phys. Rev. Lett. **48**, 1507 (1992).

[31] E. Ott, C. Grebogi and J. A. Yorke, Phys. Rev. Lett. **64**, 1196 (1990)

[32] T. Shinbrot, E. Ott, C. Grebogi and J. A. Yorke, Phys. Rev. Lett. **65**, 3215 (1990).

[33] K. Pyragas, Phy. Lett. **A170**, 421 (1992).

[34] M. K. Ali and J-Q Fang, Phys. Rev. **E55**, 5285 (1997).

[35] A. S. Pikovsky *et al*, Physica D **104** 219-238 (1997).

Report 5

Pattern Recognition Based on Stochastic Resonance in a Canonical Model

(October - March, 1999)

Contents

1 Introduction 3

2 Bifurcations in neural networks 7

3 A canonical model with bifurcations 9

4 Stochastic resonance in the canonical model 13

4.1 stochastic resonance 13

4.2 Storage capacity 19

4.3 Mixed input 20

4.4 Initial states (history) 21

5 Conclusions 23

Chapter 1

Introduction

Stochastic resonance (SR) is an interesting and important phenomenon that can dramatically affect, under right conditions, the nature of time evolution of dynamical systems [1, 2]. Recently, SR has drawn increasing attention because of its potential for application in diverse situations. A growing number of workers have the view that biological systems may actually utilize the mechanism of SR to perform nervous activities like signal processing, collective spiking and recognition [3, 4, 5, 6]. It has become a very active area of research in neuroscience and biophysics [1, 3, 7, 8].

There are experimental data [1, 3, 8] from neuroscience laboratories that provide elementary support for the existence of SR in biological systems. There are also some theoretical models [1, 4, 5, 9] designed to explain the experimental data and to understand the role of SR. Simulations based on models such as the classical Hodgkin-Huxley (HH) model [9, 10, 11], the FitzHugh-Nagumo model [7] and the two-chamber model [4, 12] show evidence of the presence of SR.

In order to investigate the new approach, we need to understand the stochastic resonance from the source, the theoretical models, the development and its present behaviors in nervous systems. In fact, we have reviewed a large number of literatures concerning the basic concept, the measurement of SR, the model of SR in the biological system and the mechanism of recognition with SR.

Stochastic resonance is a nonlinear phenomenon in which random noise optimize a system's response to a signal, which provides a role for noise in information processing in the brain. SR is now firmly established as a common phenomenon which appears in a wide variety of physical system. Figure

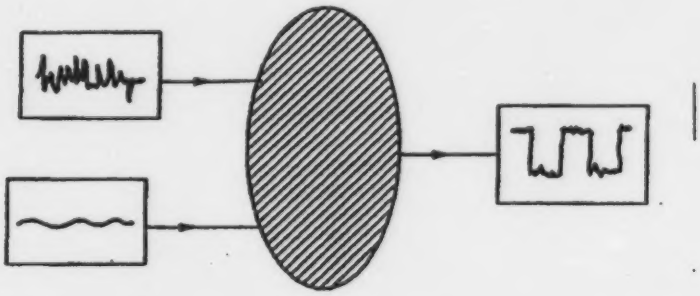

Figure 1.1: Schematic description of SR (Stochastic Resonance). The cross-hatched oval represents a black-box which receives two kinds of inputs: one weak and periodic, the other strong and random. The output is relatively regular with small fluctuations (from [1]).

1.1 gives a schematic description of SR. A system is subject to two kinds of stimuli: the first is large and random and the second is week and periodic. Typically, one expects that the output response would be largely random. However, the system displaying SR has a quite different response: the response is fairly regular with only small fluctuations. For the analysis of SR, there are theoretical models such as double-well model, bistate model, nondynamic model, etc. The nondynamic threshold model attracts the particular attention of physicists and biologists. Figure 1.2 illustrates the mechanism of the model: the state of a system is simply the sum of signal and noise inputs; whenever the sum crosses a fixed threshold in one direction it triggers a narrow pulse in the output. Some investigators have tried to apply this model to recognition [5, 6] of patterns which depends on the intensity of the random noise. This idea is actually related to the occurrence of the action potential (spiking) in individual neurons and the information encoding by the use of the timing of spiking [27, 28, 29]. In addition, a neuronal

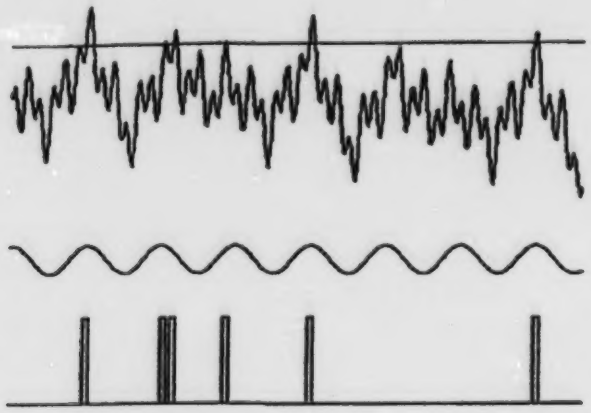

Figure 1.2: Illustration of a simple threshold model. The top trace shows the sum of random and periodic inputs; the horizontal line indicates the fixed threshold. The middle trace shows, for comparison, the periodic pae of the input. Each time total input crosses the threshold upwards, a unit pulse is generated in the output (lower trace) (from [1]).

net is composed of individual neurons whose signal is usually described by the motion of ions (potassium, sodium, calcium etc). The Hodgkin-Huxley (HH) model [10] is the classical work to study these movements. Actually, many studies are being focussed on single neuron (cell) whose behavior can be chaotic, bistable or normal [12, 30, 31, 11]. The properties of single neurons deserve attention with regard to the basic mechanism of encoding and retrieving messages. The behaviors of single neurons are related to the distribution and flow of ions (potassium, sodium, calcium etc). The solution of the Hodgkin-Huxley model in diverse range of parameters is capable of describing these properties. For example, the occurrence of spiking (action potential), the refractory mechanism can be explained by the model. After having some backgrounds about the structure of the individual cells, we can begin to consider the SR in the neuronal system.

One may ask the natural question: Is SR an artifact of the choice of the model or is it model independent? A definite answer to this question would help us ascertain the necessity of stochasticity and chaos in biosystems [13, 14, 15, 16]. One wonders if stochastic processes are essential ingredients

of the activities of the nervous system or if chaos is necessary for an animal like a rabbit to recognize odors? If chaos or noise is required for the nervous systems to function, then perhaps it may be argued that the reason for this is that a biological system, in its history of evolution, adapts to (is trained in) the noise and chaos of the environment. Later on, a chaotic or noisy stimulus helps retrieve the memory. A partial answer to the question asked above can be obtained by studying a very general model suitable to describe a large variety of dynamical situations in different areas.

Since information processing such as pattern recognition in the brain is important for biological systems, we study SR by focusing on the mechanism of recognition. Except for some limited work [5], the role of SR in this area has not yet been studied widely. The study is important not only in modelling networks, but also in possible clinical applications such as rehabilitating the injured brain. It is hoped that this work will serve as a theoretical guidance in the search for techniques to restore the injured brain and in achieving related goals of practical importance [17].

The work is aimed at studying the stochastic phenomenon using a very general model. We show the existence of SR in this general model. We also show that SR enhances pattern recognition. We believe that these results are meaningful both in understanding the brain and in applications based on neural networks.

Chapter 2

Bifurcations in neural networks

In order to describe observed activities of a system, such as the human brain, one needs mathematical models. The problem with models is that a model may describe only a specific activity and not others. One then needs many models for the activities of the same brain. The situation improves if a model is general. There are models called Canonical Models that are general and are the preferred ones. A canonical model generates the results of a family of models. It is desirable that this family is large. In this work we will use a canonical model.

Before arriving at our canonical model, we briefly discuss bifurcations in neural networks. We will reduce the model from a general equation. Suppose we have an array (network) of N neurons in which the neurons (cells) are weakly connected. The general form of the dynamics of the i-th neuron in this network is

$$\dot{x}_i = f_i(x_i, \lambda) + \epsilon g_i(x_1, \ldots, x_n), \tag{2.1}$$

where, x_i is the activity of neuron i, λ denotes a set of parameters of the system, $f_i(\cdots)$ is usually a smooth function governing the dynamics, g_i describes the connections (synapses) from other neurons to neuron i and ϵ is the strength of the connections.

The activity x_i may be the membrane potential, number of action potentials (spikes), frequency of oscillations, amount of chemical neurotransmitter (ions) or something else. In short, x_i can represent a wide variety of activities when we reduce eq. (2.1) to specific models [18]. Many models can be

abstracted into the form of eq. (2.1). For example, consider x_i representing the physiological data of membrane potentials. In this case, the postsynaptic potential size (x_i) in the hippocampal granule cells is about 0.1 mV [18, 33]. The value of ϵ depends on a threshold value (which is about 24 mV in hippocampus cells) to discharge (fire) a cell and how the cells are connected. Mathematically ϵ should be smaller than 1 for weakly connected networks. The biological analysis gives an estimation of $0.004 < \epsilon < 0.008$ [18]. Since a large number of neurons in a network are connected, the firing of some neurons may lead to the generation of an action potential. The condition for the occurrence of an action potential in a weakly connected neural network is that the membrane potential of the neuron is near the threshold value of firing the neuron, because of the connections are weak. This neurophysiological requirement is described by the mathematical eq. (2.1) as the fact that the neuron activity is near a bifurcation point. This basic relationship is actually substantiated by a theorem that, in order to make nontrivial contribution to brain activities, a neuron must be near a threshold [18]. Furthermore, if the activity of each neuron is near a hyperbolic equilibrium, the network is not interesting, since it is topologically equivalent to a linear system with trivial dynamics. All these theorems can be proved rigorously and the details can be found in [18].

As the parameters of a dynamical system are changed, the stability of its equilibrium points as well as the number of equilibrium points change. The values of these parameters at which the qualitative or topological nature of the motion changes are known as the bifurcation points. A dynamical system is thus structurally unstable at a bifurcation point. If $\lambda = \lambda_b$ corresponds to a bifurcation point, then in the neighborhood of λ_b there is a value λ_c at which the system behaves differently. Period doubling bifurcation in the logistic map to onset of chaos is well known. The relationship between neural networks and bifurcations elucidates the generality of the model. For an illuminating discussion on the canonical models for neural networks please see ref. [18]. In order to specify the relation more clearly, we will give more illustrations about bifurcation and reduce eq. (2.1) to a usable model.

Chapter 3

A canonical model with bifurcations

In neuroscience, there are many models to describe the same brain structure. All these models may be valid to illustrate a specific phenomenon in the brain. Unfortunately, various models may give different results when they are applied to study the brain beyond that phenomenon. Therefore, we need to "extract" a model that is capable of giving results observed in a family of models. We need to reduce the family of models to a simple model. When the model is modified by adding more parameters and variables, similar results should hold. Such a model is called a canonical model. If there is certain dynamics characteristic of the family, a canonical model is capable of representing all members of the family in this regard. A canonical model possesses characters of simplicity (it is simpler than any dynamic system in the family) and universality (it captures some essentials of every member of the family). Therefore, it is important for us to resort to canonical models, since in many cases of neuroscience, we usually have only restricted knowledge. A correctly established canonical model should give us more information than what is currently known. Technically, canonical models can be obtained by using normal form and invariant manifold theories [18]. We do not discuss them in detail in this paper.

As discussed briefly in chapter 2, in a weakly connected neural network, brain activities occur only when the neurons are near thresholds. In fact, this is the

starting point ("fundamental theorem " called in [18]) for studies of weakly connected networks. It is known that different bifurcations can be used in various dynamic systems. For example, saddle-node is a basic approach to illustrate many behaviors such as stability in additive model, Wilson-Cowan model, Cusp bifurcation is necessary to describe behavior with attractor, while Andronof bifurcation plays key role to study the oscillator/limit circle. For different bifurcations, eq. (2.1) will be reduced to different forms. In this paper we concentrate on pattern recognition as we've done before [20, 21, 22]. For recognition we need a model that has either GAS (globally asymptotically stable) or MA (multiple attractors, even chaotic attractors). In order to perform the task, we consider cusp bifurcation in our model. In what follows, first we reduce eq. (2.1) to a canonical model and then we show that this model can be used to recognize patterns.

The cusp bifurcation of a dynamical system $\dot{x}_i = f_i(x_i, \lambda)$ is described as an equilibrium point $x_i = 0$ for $\lambda = 0$, i.e. $f_i(0,0) = 0$, if at this point the following conditions are satisfied:

$$\begin{aligned} \frac{\partial f_i}{\partial x_i} &= 0 \quad \text{(nonhyperbolicity)} \\ \frac{\partial^2 f_i}{\partial x_i^2} &= 0 \quad \text{(no quadratic term) and} \\ \frac{\partial^3 f_i}{\partial x_i^3} &\neq 0 \quad \text{(presence of cubic term).} \end{aligned} \tag{3.1}$$

With above equations, it can be shown [18] that the dynamics of *any* weakly connected neural network at a cusp bifurcation is governed by the canonical model

$$x_i' = r_i + b_i x_i - x_i^3 + \sum_{j=1}^{n} c_{ij} x_j, i = 1, \ldots, n, \tag{3.2}$$

In (3.2) $x_i' = \frac{dx_i}{d\tau}$ with τ being the "slow time" ($\tau = \epsilon t$, t is time), r_i, b_i and c_{ij} are parameters (real variables) whose meanings will be illustrated below. Why are we interested in this canonical model? We have to investigate basic points about the model before we utilize it for pattern recognition. First we check the global behavior of the model. The global property tells us that

this model is bounded. This means that the system will not move apart infinitely from an equilibrium point, in other words, the system being far from equilibrium will move in the direction towards the equilibrium. Therefore, this global property guarantees that for any initial condition we may find an asymptotic behavior that is necessary for the system to perform recognition.

The parameter r_i represents the input from receptors that get the information from the world outside the network. When the input is strong (r_i is much larger than b_i and c_{ij}), the system has an unique stable equilibrium that is determined entirely by r_i: $r_i > 0 (< 0)$ leads to $x_i > 0 (< 0)$, i.e. neuron i is exited ($x_i > 0$) or inhibited $x_i < 0$) [18].

The parameter b_i describes the psychological state in the sense it affects the response of the system to the external input (represented by r_i). When b_i takes very large values, the system has equilibria at $y_i = 0$ for $b_i < 0$ or $y_i = \pm\sqrt{b_i}$ for $b_i > 0$ with $y_i = \sqrt{min_i|b_i|}$. These equilibria are stable (unstable) when the above conditions are satisfied (violated). Later in this article, we will see that b_i is the bifurcation parameter determining the occurrence of a saddle-node bifurcation [18]. This bifurcation plays the key role in performing the recognition.

The parameter c_{ij} describe the strengths of connections (synapses) that will be determined by a learning rule that we suppose to be the Hebbian rule (discussed later). Anyway, the various values of these parameters give different behaviors and equilibria that make the model applicable in investigating the pattern recognition: the canonical model can serve as a network with a global asymptotically stable attractor or with multiple attractors, depending on the parameters r_i, b_i and c_{ij}. Both types of these attractors possess equilibria that are necessary for recognition.

We assume that the synapses are determined by the Hebbian rule, i.e. if the number of p randomly orthogonal memories stored in the system are $\xi = \{\xi_i^\mu = \pm 1$ with $i = 1, \ldots, N$ and $\mu = 1, \ldots, p\}$, the synapses are

$$c_{ij} = \frac{c}{N} \sum_{\mu}^{p} \xi_i^\mu \xi_j^\mu. \tag{3.3}$$

This is the most commonly used rule in modelling neural networks.

Why do we introduce cusp for pattern recognition? As discussed above, the model (3.2) obtained in this way has GA, GAS and other type attractors that are necessary for recognition, depending on the parameters. The discussions about the existence of MA will be given later. Here we just give the statement that model (3.2) has one equilibrium at $x_i = 0$ for $b \leq -c$ and two equilibria at $x_i = \pm\sqrt{b + c}\xi_i^\mu$ for $b > c$ that are related to patterns. The two equilibria (attractors) play important roles in recognition.

The canonocal model (3.2) is general for it is obtained with simple known connections between neurophysiology and bifurcations. We now return to the problem proposed in our introduction – stochastic resonance in a universal meaning. Since pattern recognition is one of the most important function of brains, we study SR by focusing on this function in the canonical model.

Chapter 4

Stochastic resonance in the canonical model

4.1 stochastic resonance

Our work is based on the canonical model of Eq.(3.2). The variable x_i is the state of neuron i. The neuron is considered to be in an excited state if $x_i > 0$ and in an inhibited state if $x_i < 0$. The recognition process refers to a specified relation between a pattern, e.g. ξ^1, and the state $x(t)$ of the network at time t. Both ξ^1 and $x(t)$ are N-dimensional randomly orthogonal vectors with components $\xi^1 = (\xi_1^1, \xi_2^1, \ldots, \xi_N^1)$ and $x(t) = (x_1(t), x_2(t), \ldots, x_N(t)$. Usually, a distance is introduced in the space of states to describe the relation in pattern recognition [20, 21, 22],

$$d(t) = \sum_i^N |\xi_i^1 - x_i(t)|. \tag{4.1}$$

As the system evolves in time from the initial state $x(0)$, the distance $d(t)$ should become sufficiently small if the network has succeeded in recognizing the pattern ξ^1. In this work, we introduce a new quantity that is more convenient to describe recognition by the canonical model. Recall that ξ_i^1 takes the value $+1$ or -1. Because a neuron is either excited ($x_i(t) > 0$) or inhibited ($x_i(t) < 0$), $x_i(t)$ should synchronize with ξ_i^1, *if* the network is capable of recognizing this pattern. In other words, synchronization between the state and the pattern implies recognition. Keeping this in mind, we define

a probability

$$pr(t) = \langle \frac{1}{N} \sum_{i}^{N} s(x_i(t)\xi_i^1) \rangle_\xi \tag{4.2}$$

with $s(y) = 1(0)$ for $y > 0 (< 0)$ to describe the probability with which the state is synchronized with pattern ξ^1 ($x_i(t)$ and ξ_i^1 have the same sign). Here $\langle\rangle_\xi$ means the ensemble average over patterns that are encoded in the synapses. If ξ^1 can be recognized, $pr(t)$ will approach the value 1 as the system evolves in time from an initial state $x(0)$.

Because the dynamics of Eq. (3.2) depends on the parameters and the initial state $x(0)$, we begin our study with the trial values

$$\begin{aligned} N &= 100 \\ p &= 15 \\ r_i &= 0.3\xi_i^1, \quad i = 1, \ldots, N \\ b_i &= 1.5, \quad i = 1, \ldots, N, \\ c &= 0.5 \\ \epsilon &= 0.5 \end{aligned} \tag{4.3}$$

and define $x(0)$ by flipping ξ^1 with probability p_f and keeping ξ^1 unchanged with probability $1 - p_f$. i.e.

$$p(x(0)) = p_f \delta_{x(0),-\xi} + (1 - p_f)\delta_{x(0),\xi} \tag{4.4}$$

which gives the probability for $x(0)$ to take ξ^1 and $-\xi^1$. Sometimes, $x(0)$ is called the history of the net for it must be defined *before* the dynamics starts. Psychologically, it corresponds to the memory we have processed before we begin a new activity. Its effect on the recognition is obvious and will be discussed later. As mentioned above, N is the number of neurons in the net, p is the number of patterns, the r_i are external inputs from receptors, the b_i are bifurcation parameters, c is the strength of synapses and ϵ is the coupling strength. o begin with, we chose the relatively large value, $\epsilon = 0.5$. The effect of a typical and much smaller value 0.005 of ϵ (see Chapter 2) will be discussed later in this article. The parameter values given in Eq.(4.3) will be modified and the effects of the modifications will be studied.

Having chosen the parameter values and the initial state $x(0)$, Eq. (3.2) is solved by a fourth order Runge-Kutta method with step size 0.1. We have solved Eq.(3.2) with and without noise. First, we discuss our results obtained without any additive noise.

Figures 4.1(a) and (b) give time evolution of the probability $pr(t)$ and the distance $d(t)$, respectively.

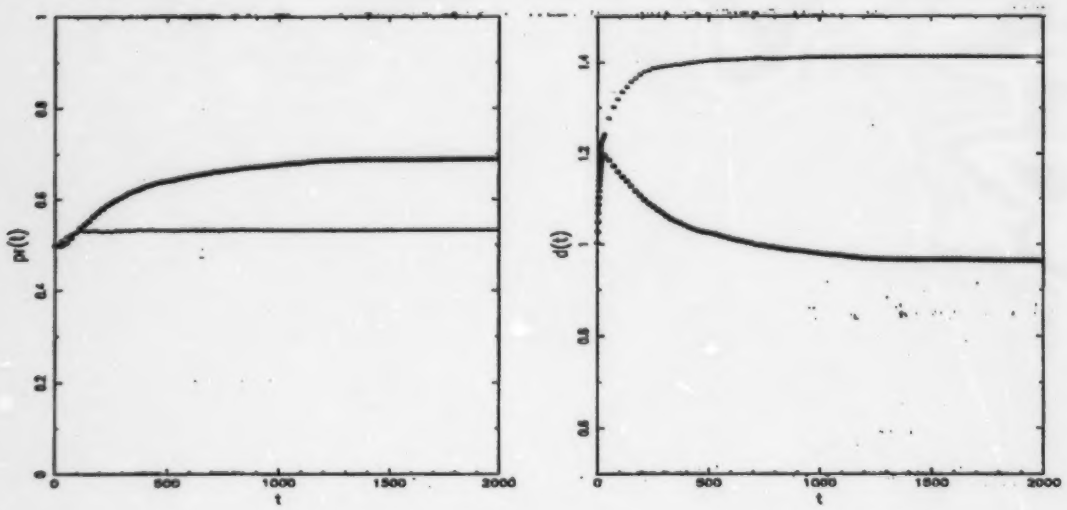

Figure 4.1: (a) The probability of synchronization $pr(t)$ vs. time t and (b) the distance $d(t)$ vs. time t for $r_i = 0.1$ (stars) and $r_i = 0.3$ (circles). No stochastic term is added.

Because the initial state $x(0)$ is obtained by flipping the pattern ξ^1 in a random way with probability $p_f = 0.5$ (see Eq, (4.4)), the initial value of the probability $pr(0) = 0.5$ and the initial distance $d(0) = 1$. The stars and the circles in Fig. 4.1 represent the data for $|r_i| = 0.1$ and 0.3 respectively (see Eq. (4.3)). Larger $|r_i|$ means injection of stronger information about the stored pattern(s) into the system through its sensor neurons. It is observed that a larger $|r_i|$ leads to a larger probability of synchronization of the state $x(t)$ with the pattern ξ^1 and a smaller distance $d(t)$. For example, after

2000 time steps, $pr(t)$ for $|r_i| = 0.3$ is 0.69 while $pr(t) = 0.53$ for $|r_i| = 0.1$. Similarly, $d(t) = 0.96$ for $|r_i| = 0.3$ and $d(t) = 1.2$ for $|r_i| = 0.1$. ¿From these values of $pr(t)$ and $d(t)$, we cannot say that the net is able to perform recognition, since the values of $pr(t)$ and $d(t)$ do not satisfy the criterion for recognition as discussed above. Note that the distance $d(t)$ will never be zero even if the input is the pattern ξ^1 itself i.e. $|r_i| = 1$ unless the initial state $x(0) = \xi^1$. This is again the effect of the history mentioned above. And this is the reason why we prefer to use $pr(t)$ as the criterion for recognition. If we use $d(t)$ to judge recognition, we have to subtract the "background" because of the deflection of $x(0)$ from ξ^1. We return to this point later.

The picture regarding pattern recognition by our neural net changes dramatically when we add noise to Eq. (3.2). Let us write Eq. (3.2) with a noise term as

$$x_i' = r_i + b_i x_i - x_i^3 + \sum_{j=1}^{n} c_{ij} x_j + \eta_i(t), \; i = 1, \ldots, n \tag{4.5}$$

where $\eta_i(t)$ is the gaussian white noise described by

$$\begin{aligned} \langle \eta_i(t) \rangle &= 0 \\ \langle \eta_i(t) \eta_i(t') \rangle &= I\delta(t - t'), \end{aligned} \tag{4.6}$$

Here I is the intensity of stochastic noise and $\langle\rangle$ denotes the ensemble average over the random noise. With the parameter values and initial state of Eqs. (4.3) and (4.4), we study the performance of our net with regard to pattern recognition by varying I within a range of values. The results are shown in Figure 4.2(a). The probability of synchronization between the state and pattern after 2000 time steps, varies with the noise intensity I. In a specific (optimal) range of I, $pr(2000)$ reaches or is very close to 1 satisfying an absolute reliable criterion for complete synchronization and it means that the pattern is recognized. This astonishing result gives hard evidence that the performance in recognition, an important feature of information processing in neural networks, is enhanced by introducing noise into the network. As our model is canonical and prototypical of a large family of models, we believe that this behavior is generic to this family. In comparison to the result $pr(t = 2000) = 0.69$ for $I = 0$, we have $pr(t = 2000) = 0.99$ for $I = 10$! Beyond the optimal range, $pr(t = 2000)$ takes different values. When the

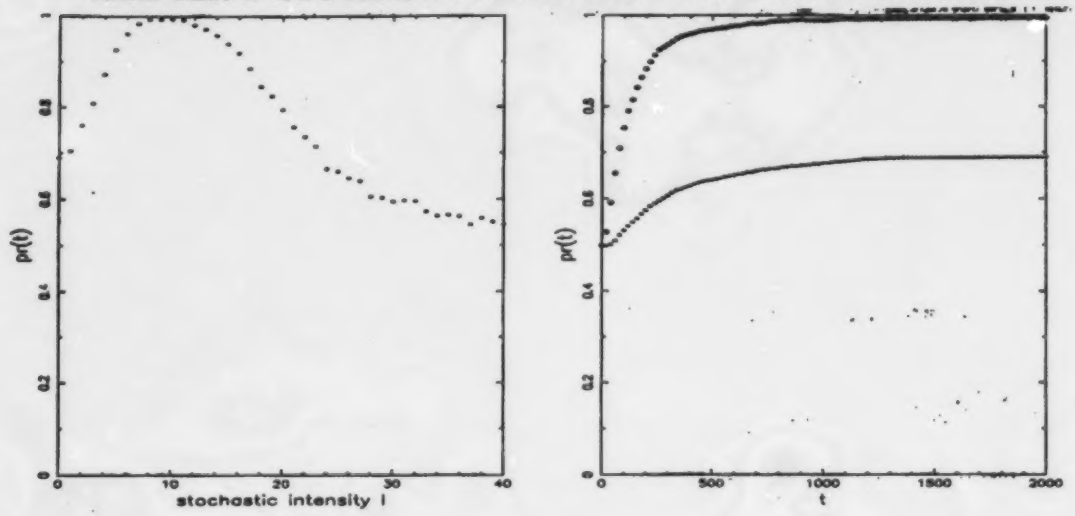

Figure 4.2: (a) The probability of synchronization $pr(t = 2000)$ vs. noise intensity I and (b) $pr(t)$ vs. time t with (circles) and without (stars) noise. Parameter set eq (4.3) and (4.4) are used.

noise is too strong, the performance is degraded because very strong noise undermines the effect of sensory inputs for any given $|r_i|$. For the purpose of comparison, we present in Fig 4.2(b) the time evolutions of pr with and without the additive noise. Again, we observe the existence of SR from the dependance of pr on t. The plot with stars (without noise) is lower than that with circles (noise level $I = 10$).

The enhancement of recognition in the presence of noise at an appropriate range of noise level is a manifestation of stochastic resonance. In order to explain the phenomenon, we first set $r_i = 0$ and introduce the transformation [18]

$$x_i = y_\mu \xi_i^\mu \tag{4.7}$$

that changes Eq.(3.2) to

$$y'_\mu = (b + c)y_\mu - y_\mu^3. \tag{4.8}$$

where $y' = \frac{dy}{d\tau}$.

Equation (4.8) has equilibria at: (1) $y_\mu = 0$ when $b < -c$ and (2) $y_\mu = 0, \pm\sqrt{b+c}$ for $b > -c$. Therefore, the equilibria of (3.2) are at $x_i = 0$ for $b < -c$ and at $x_i = 0, x_i = \pm\sqrt{b+c}\xi_i^\mu$ for $b > -c$. The stability analysis shows [18] that the equilibria at $x_i = 0, x_i = \pm\sqrt{b+c}\xi_i^\mu$ are stable. Therefore, around $b = -c$ there is a bifurcation: when $b < -c$, only one trial equilibrium $x_i = 0$ exists; as b increases through $-c$, the equilibrium at $x_i = 0$ for $b < -c$ bifurcates to two new equilibrium points $x_i = \pm\sqrt{b+c}\xi_i^\mu$. Note that these equilibria are related to the pattern ξ^μ. We can say that the pattern ξ^μ is recognized when we find the system near these points. If the system is perturbed by the external inputs $|r_i| \neq 0$ and the noise, neuron i may shift between these two equilibria. This is similar to transitions between the two wells of a "double well" problem that is known to serve as the classic model for the analysis of stochastic resonance [1, 26]. If the inputs $|r_i|$ are not too strong (a common situation in recognition), the input signals alone cannot cause a transition between the two states by overriding the barrier between them. Crossing the barrier is nothing but crossing a threshold. "Aided" by the noise, the transition occurs. The frequency depends on the strength of the noise. Another important and interesting question is: Why does the transition favor synchronization with the correct pattern. Usually, SR does not suggest that the transition will happen in favor of a particular direction. The double-well model says that the probability of transition from one state (well) to the other does not depend on the well. The answer lies in the encoding of synapses. We are dealing with an intelligent system with memory and not with a general system. The learning rule provides information about which patterns are learned and are stored. Therefore, the system behaves as though it "knew" how and where to jump. The explanation of the results in Fig.4.2 combines the generic feature of SR and the character of intelligent system.

We now address the following questions: (1) Is this SR of synchronization connected to the memory load (storage capacity)? (2) Does this SR still exist when the inputs from the receptors are mixtures of more than one pattern? (3) What is the effect of the initial state (history)? (4) If we change the values of the parameters, is there any change in the SR? and (5) Is there a size effect? In the next chapter we attempt to answer these questions.

4.2 Storage capacity

The ability of a neural network to recognize patterns is intimately related to its storage capacity [23, 24, 25, 20]. We now turn to the storage capacity of our canonical network. In Fig. 4.2, we present the case of $p = 15$ with the net size $N = 100$. Figure 4.3(a) contains curves corresponding to (1) $p = 15$ (stars), (2) $p = 30$ (circles), (3) $p = 50$ (crosses) and (4) $p = 100$ (squares). It can be seen from these figures that the probability of syn-

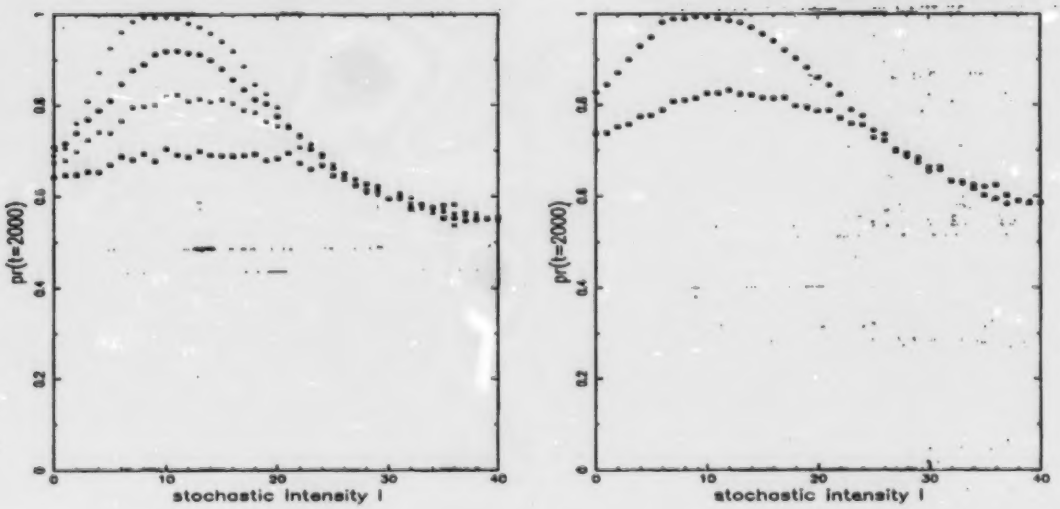

Figure 4.3: (a) The probability of synchronization $pr(t = 2000)$ vs. noise intensity I for $p = 15$ (stars),$p = 30$ (circles), $p = 50$ (crosses) and $p = 100$ (squares) and (b) $pr(t = 2000)$ vs. I for $p = 30$ (circles) and $p = 100$ (squares) with $r_i = 0.5$. Other parameter set are 4.3 and 4.4.

chronization decreases with increasing p. This behavior is typical of other well-know models [23, 25]. It is interesting to note that SR is present even though $pr(t)$ decreases as p increases. Around $I = 10$, the probabilities reach maximum values that are obviously larger than $pr(t)$ obtained without noise. When the input includes more information on the pattern, the recognition is improved. Figure 4.3(b) shows the results for $p = 30$ (circles) and $p = 100$ (squares) by increasing $|r_i|$ to 0.5, while the other parameters are the same

as those used for Fig.4.3. The probability is clearly upgraded.

4.3 Mixed input

When the receptors are fed with information about more than one pattern *simultaneously*, the network can still perform recognition in terms of strengths of various patterns? For the purpose of illustration. we consider the case of

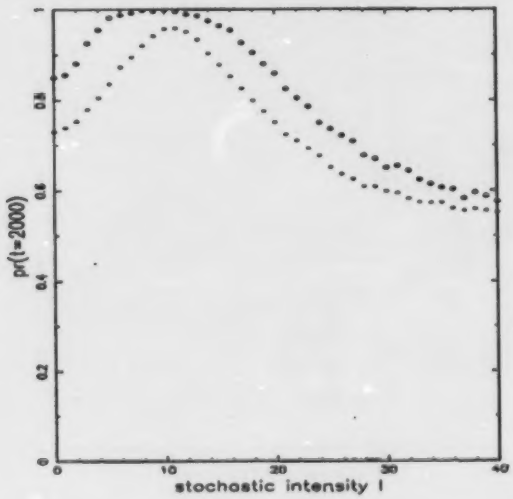

Figure 4.4: The probability of synchronization $pr(t = 2000)$ vs. noise intensity I for $r_i^1 = 0.3$, $r_i^2 = 0.2$ (stars) $r_i^1 = 0.5$, $r_i^2 = 0.2$ (circles).

two patterns, e.g. ξ^1 and ξ^2. Now the inputs have the form

$$r_i = r_i^1 \xi_i^1 + r_i^2 \xi_i^2, \quad i = 1, \ldots, N, \tag{4.9}$$

where r_i^1 and r_i^2 are the strengths of ξ_i^1 and ξ_i^2, respectively. Figure 4.4 shows the probability of synchronization vs. noise intensity for two choices of r_i^1 and r_i^2. The results depend on the contrast between the patterns. The probability for larger contrast (circles, $r_i^1 = 0.5$ and $r_i^2 = 0.2$) is larger than that

for smaller contrast (stars, $r_i^1 = 0.3$ and $r_i^2 = 0.2$). This is in agreement with the discussions about the so-called bistablity [18]. For both cases we observe SR. It is very important to point out that SR filters out completely the ambiguous information about ξ^2. Without noise ($I = 0$), the ambiguous information included in the input will stay in the dynamics and therefore, pr will never approach 1. This fact has been observed and discussed in other important models like the Hopfield model [24] and the Gardner model [20], where such input leads to metastable (spurious) states that traps the dynamics and therefore, degrades the ability of recognition. From Fig.4.4 we see that even for larger contrast pr is only 0.85 for $I = 0$, while it is 0.997 for $I = 10$. This should be stressed as another effect of SR. SR helps the system remove the less important information (denoted by smaller weight in eq. (4.9)). In comparison with the discussions in ref. ([18]), where the quasi-static process is introduced to discuss the recognition, the recognition of ambiguous information is performed by SR.

4.4 Initial states (history)

The initial state $x(0)$ corresponding to the history of the system influences the results of recognition. As an example, we studied the effect with the three choices $p_f = 0.1$, 0.3 and 0.5. The initial states are obtained by flipping pattern ξ^1 with probability p_f according to Eq. (4.4). Figure 4.5 shows the result for the three initial states when the number of 50 patterns are stored. When the initial state is closer to the pattern (p_f takes smaller value), the probability is larger and the recognition is thus better. Note that $p_f = 0.5$ means the initial state is randomly orthogonal to the pattern, since a component of a pattern takes either 1 or -1.

As for the size effect, we did not observe any difference in qualitative behaviors when we chose $N = 100$ and $N = 500$. We also used small ϵ ($\epsilon = 0.005$) but did not observe structural or qualitative changes in our results. Of course large N and/or small ϵ means longer CPU time.

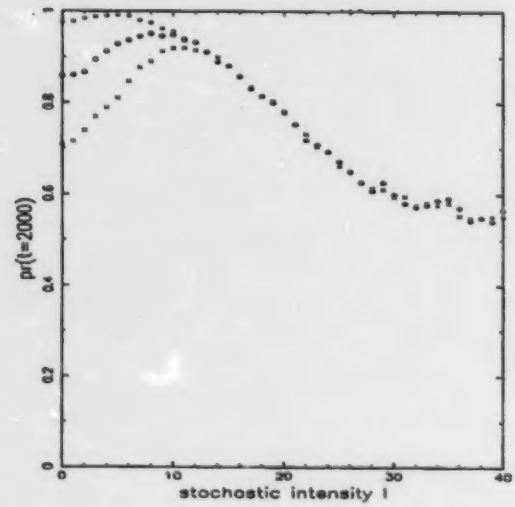

Figure 4.5: The probability of synchronization $pr(t = 2000)$ vs. noise intensity I for three initial states $p_f = 0.1$ (stars), 0.3 (circles) and 0.5 (crosses).

Chapter 5

Conclusions

Pattern recognition, an important aspect of information processing by intelligent systems, has been discussed with a canonical model. This model is derived using simple assumptions based on the well-known relation between neurophysiology and bifurcation theories. Being canonical, the model is capable of generating results that are characteristic of a large family of models near the bifurcations. Therefore, it is able of describing a large number of phenomena. The phenomenon of SR has been found to enhance pattern recognition. Our work on this canonical model suggests that SR is more universal than current beliefs in systems near thresholds. For the network of this work, SR helps not only to synchronize the time evolved state with the patterns, but also filters out the ambiguous input information in terms of the strengths. The work is specially meaningful in following two ways:

1. SR is verified to exist in a more universal sense.

2. SR is necessary for the system to perform recognition.

We have also discussed SR in light of the storage capacity, the initial conditions and the ambiguous (mixed) input. All these results establish a complete set of SR and pattern recognition in the canonical model which can describe a lot of phenomena.

Bibliography

[1] K. Wissenfield and F. Jaramillo, "Minireview of stochastic resonance", Chaos **8**, 539 (1998).

[2] L. Gammaitoni, P. Häggi and F. Jung, Rev. Mod. Phys. **70**, 223 (1998)

[3] B. J. Gluckman, T. I. Netoff, E. J. Neel, W. L. Ditto, M. L. Spano, and S. J. Schiff, "Stochastic resonance in a neuronal network from mammalian brain", Phys. Rev. Lett. **77**, 4098 (1996).

[4] B. J. Gluckman, P. So, T. I. Netoff, M. L. Spano and S. J. Schiff, "Stochastic resonance in mammalian neuronal network", Chaos, **8**, 588 (1998).

[5] E. Simonotto, M. Riani, C. Seifer, M. Robert, J. Twitty, and F. Moss, "Visual perception i of stochastic resonance", Phys. Rev. lett. **78**, 1186 (1997).

[6] M. Riani and E. Simonotto, "Stochastic resonance in the perceptional interpretation of ambiguous figures: a neural network model", Phys. Rev. Lett. **72**, 3120 (1993).

[7] J.J.Collins, C. C. Chow and T. T. Imhoff, "Stochastic resonance without tuning", Nature **376**, 236 (1995).

[8] X. Pei and F. Moss, "Characterization of low-dimensional dynamics in the crayfish caudal photoreceptor", Nature **379**, 618 (1996)

[9] X. Pei, L. Wilkens and F. Moss, "Noise-mediated spike timing precision from aperiodic stimulus in an array of Hodgkin-Huxley-type neurons", Phys. Rev. Lett. **77**, 4679 (1996).

[10] A. L. Hodgkin, *The conduction of the nervous impulse*, (Charles C Thomas · Publisher, Springfield, Illinois, 1964).

[11] H. C. Tuckwell, *Introduction to Theoretical Neurobiology*, Vol. 2, (Cambridge U. P., Cambridge, 1988)

[12] V. Booth and J. Rinzel, "A minimal, Compartmental model for a dendritic origin of bistable of motoneuron firing patterns", Journal of Computational and Neuroscience, **2**, 299 (1995).

[13] C. A. Skarda and W. J. Freeman, Behav. Brain Sci. **10**, 161 (1987); W. J. Freeman, Biol. Cybernetics, **56**, 139 (1987).

[14] P. So, J. T. Francis, T. I. Netoof, B. J. Gluckman and S. Schiff, "Periodic Orbits: A new language for neuronal dynamics", Biophysical Journal, **74**, 2776 (1998).

[15] I. Tsuda, Neural Network **5**, 313 (1992).

[16] K. Aihara, T. Takabe and M. Toyoda, Phys. Lett. **A114**, 333 (1990).

[17] J. Glanz, Science **277**, 1758; *ibid.* 1759 (1997).

[18] F. C. Hoppensteadt and E. M. Izhikevich, *Weakly Connected Neural Networks*, (Springer-Verlag, New York, 1997).

[19] J. A. Anderson, *An introduction to Neural Networks*, (MIT, Cambridge, 1995).

[20] Z. Tan and M. K. Ali, Phys. Rev. **E57**, R3739 (1998);

[21] Z. Tan and M. K. Ali, Phys. Rev. **E58**, 3649 (1998);

[22] Z. Tan, B. S. Hepburn, C, Tucker and M. K. Ali, Discrete Dynamics in Nature and Society **2**, 243 (1998).

[23] J. J. Hopfield, Proc. Nat. Acad. Sci. USA Vol. **79**, 2554 (1982).

[24] D. J. Amit, H. Gutfreund and H. Sompolinsky, Phys. Rev. **A32**, 1007 (1985).

[25] E. Gardner. J. Phys. **A21**, 257 (1988)

[26] A. Longtin, A. Bulsara and F. Moss, Phy. Rev. Lett. **67**, 656 (1991).

[27] J. J. Hopfield, "Pattern recogniton computation using action potential timing for stimulus representation", Nature **376** 33 (1995).

[28] D. W. Tank and J. J. Hopfield, "Neural computation by concentrating information in time", Proc. Natl. Acad. Sci. **84**, 1896 (1987).

[29] C. E. Carr and M. Konishi, "Axonal delay lines for time measurment in the owl's brainstem", Proc. Natl. Acad. Sci. **85**, 8311 (1988).

[30] X.-J. Wang and J. Rinzel, "Oscillatory and bursting properties of neurons", *Handbook of Brain Theory and Neural Networks*, ed. M. Arbib, (Cambridge, Mit Press 1995).

[31] J. M. Buhmann, "Oscillatory Associative Memories", *Handbook of Brain Theory and Neural Networks*, ed. M. Arbib, (Cambridge, Mit Press 1995).

[32] M. Yoshioka and M. Shiino, "Associative memory based on synchronized firing of spiking neurons with time-delayed interactions", Phys. Rev. E **58**, 3628 (1998).

[33] J. A. Anderson, *An introduction to Neural Networks*, (Mit Press, Cambridge, 1995).

[34] P. So, E. Ott, S. J. Schiff, D. T. Kaplan, T Sauer, and C. Grebogi, "Detecting unstable periodic orbits in chaotic experimental data", Phy. Rev. Lett. **76**, 4705 (1996).

[35] S. J. Schiff, K. Jerger, D. H. Duong, T. Chang, M. L. Spano and W. L. Ditto, "Controlling chaos in the brain", Nature, **370**, 615 (1994).

[36] E. Tziperman and H. Scher, "Controlling spatiotemporal chaos in a realistic El Niño prediction model", Phys. Rev. Lett. **79**, 1034 (1997).

Reprinted from

PHYSICAL REVIEW
E

STATISTICAL PHYSICS, PLASMAS, FLUIDS, AND RELATED INTERDISCIPLINARY TOPICS

APRIL 1998

Mixed states in a neural network model

Z. Tan and M. K. Ali
Department of Physics, University of Lethbridge, Lethbridge, Canada T1K 3M4

RC R3739-R3741

Published by
THE AMERICAN PHYSICAL SOCIETY
through the
American Institute of Physics

Volume 57 Third Series Number 4

Mixed states in a neural network model

Z. Tan and M. K. Ali
Department of Physics, University of Lethbridge, Lethbridge, Canada T1K 3M4
(Received 23 October 1997)

Mixed states that are symmetrically related to more than one pattern are shown to exist in the Gardner model. Their retrievability and stability depend on the storage capacity of the network. We show that, under certain conditions, these states are stable and they have properties similar to those of the mixed states of the Hopfield model. [S1063-651X(98)50504-1]

PACS number(s): 87.10.+e, 75.10.Nr, 75.50.Lk

The Hopfield model [1,2] and the Gardner model [3] are widely used in neural networks. The synapses in the two models are generated differently. The synapses in the Hopfield model are determined by the Hebbian rule, while the synapses in the Gardner model are found by maximizing the storage capacity (see below) under the requirement of a desired input-output relation. Because of a number of appealing features such as flexibility in statistical analyses and large storage capacity, the Gardner model has become one of the important models. Both models have associative memory: the models are capable of retrieving the stored patterns by a stimulus of partial information about the patterns. The patterns in these models are stable states in the state space. In the Hopfield model there are additional states that are attractors but different from the stored patterns. Amit, Gutfreund, and Sompolinsky [2] have shown that there are metastable mixed states that are simultaneously related to more than one stored pattern. These metastable states are attractors and influence the dynamics of the network. For the Gardner model, one may ask: (1) Are there such mixed states? (2) If there are such states, are they stable? (3) How are these states related to the storage capacity of the model? The aim of this paper is to address these questions.

Since a full analytical treatment turns out to be impossible, we resort to numerical methods. However, the evolution through the first time step has been treated analytically by means of a dilution technique [4]. Our numerical and analytical results show that mixed states exist in the Gardner model and that they have properties similar to those of the Hopfield model [2].

In the Gardner model with N neurons, p patterns are remembered by the network when the condition

$$\frac{1}{\sqrt{N}}\xi_i^\mu \sum_{j\neq i} J_{ij}\xi_j^\mu > K \geq 0 \tag{1}$$

is satisfied for all $i, i=1,\ldots,N$ and all μ, $\mu=1,\ldots,p$. A symmetrical mixed state has the same overlap (Hamming distance) with several patterns. In the Hopfield model, the mixed state can be obtained analytically from a mean-field equation. However, in the Gardner model, no analytical approach is known for obtaining such a state. Therefore we find the mixed state ζ from an initial state ξ' that is defined by the probability

$$p(\xi_i') = \frac{1}{n}\sum_{\mu=1}^{n} \delta(\xi_i' - \xi_i^\mu), \quad i=1,\ldots,N, 1\leq n\leq p. \tag{2}$$

Equation (2) means that ξ' is related to n patterns with the same probability $1/n$. The dynamics, when initiated with ξ', evolves to an equilibrium state that is the mixed state ζ for the Gardner model. In Eq. (2), $n=1$ corresponds to a non-mixed state, which is the usual case discussed in [3,5].

For studying the behaviors of states that lie in close proximity to ξ', we consider the time evolution of a typical state $S(t)$ that starts at t close to ξ'. Its evolution is studied by considering the Hamming distance (overlap) between one of the n patterns (e.g., ξ^1) and the state $S(t)$, $M(t) = (1/N)\Sigma_i^N \xi_i^1 S_i(t)$. It may be pointed out that $M(t)=1/n$ when $S(t)=\xi'$. The overlap after one time step is obtained by generalizing the evaluation of [5,6]

$$M(t+1) = \int_{-\infty}^{+\infty} P_n(\Lambda)\,\mathrm{erf}\left(\frac{nM(t)\Lambda}{\sqrt{2\{1-[nM(t)]^2\}}}\right) d\Lambda \tag{3}$$

with

$$P_n(\Lambda) = \left\langle \delta\left(\Lambda - \frac{1}{\sqrt{N}}\sum_{j\neq i} \xi_i^1 J_{ij}\xi_j'\right)\right\rangle_{\xi',\xi}, \tag{4}$$

where $\langle\cdots\rangle_{\xi',\xi}$ is the average over ξ' and ξ. After taking the average over ξ', the sum in Eq. (4) can be expressed as

$$\sum_{\mu=1}^{n} \frac{1}{\sqrt{N}}\xi_i^1 \sum_{j\neq i}^{(\mu)} J_{ij}\xi_j^\mu. \tag{5}$$

where, $\Sigma_{j\neq i}^{(\mu)}$ contains only the terms related to ξ^μ and totally it has N/n such terms. The sum $\Sigma_{j\neq i}^{(\mu)}$ can be understood as the corresponding quantity of a diluted network obtained by cutting randomly $N[1-1/(n)]$ synapses connecting to neuron i (in the fully connected network, the neuron i has $N-1$ synapses). Because $\xi_i^\mu = \pm 1$, ξ_i^1 in Eq. (5) can be replaced by ξ_i^μ with possible changes of sign, after considering all possible relations among $\xi_i^1, \xi_i^2, \ldots, \xi_i^n$. Hence, the sum in Eq. (5) has the form of $X_1 \pm X_2 \pm \cdots \pm X_n$ with

$$X_\mu = \frac{1}{\sqrt{N}}\xi_i^\mu \sum_{j\neq i}^{(\mu)} J_{ij}\xi_j^\mu, \quad \mu=1,\ldots,n. \tag{6}$$

The relation that brings l minus signs to the sum is described by a random walk procedure in which there are l steps walking toward the negative direction, after a movement of total $n-1$ steps. Equation (4) thus reads

 © 1998 The American Physical Society

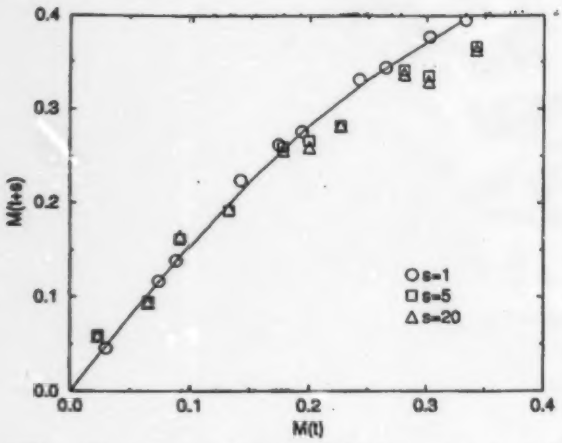

FIG. 1. The evolution of overlap $M(t)$ for $n=3$ at $K=2$. The solid line is the result of analytical treatment.

$$P_n(\Lambda)=\frac{1}{2^{n-1}}\sum_{l=0}^{n-1}\binom{n-1}{l}\langle\delta[\Lambda-f_l(X_1,X_2,\ldots,X_n)]\rangle_\xi, \tag{7}$$

where, $f_l(X_1,X_2,\ldots,X_n)$ is the algebraic sum $X_1\pm X_2\pm\ldots\pm X_n$ with l minus signs. The distribution $\langle\delta[\Lambda-f_l(X_1,X_2,\ldots,X_n)]\rangle_\xi$ can be obtained by

$$\begin{aligned}&\langle\delta[\Lambda-f_l(X_1,X_2,\ldots,X_n)]\rangle_\xi\\&=\int_{-\infty}^{+\infty}\left[\prod_{\mu=1}^{n}dX_\mu P_\mu(X_\mu)\right]\\&\quad\times\delta[\Lambda-f_l(X_1,X_2,\ldots,X_n)],\end{aligned} \tag{8}$$

where $P_\mu(X_\mu)$ is just the distribution of X_μ in a quenched diluted network [4], where $N-(N/n)$ out of the total $N-1$ synapses have been removed randomly from every neuron.

Figure 1 gives the overlap $M(t+s)$ as a function of $M(t)$ after different time steps s for $n=3$. The solid line is the analytical result for $s=1$, while the circles, the squares and the triangles are the results of simulation after 1, 5, and 20 steps, respectively. The analytical and numerical overlaps after one step agree well. The overlap after one time step plays an important role in a saturated network [5]. Starting from ξ^l, after a certain number of time steps, the equilibrium state ζ is obtained when there is no more change in the overlap. Table I shows the overlap M_3 [$M_3=\lim_{s\to\infty}M(t+s)$] of the mixed state for $n=3$ and various K. For $n=2,4,5,\ldots$, we obtained similar results.

In addition, we find an asymptotic maximum value 0.5 of M_3 for large K. In fact, such values (denoted by M_n^0) exist for all n. After some algebra to simplify the distribution of Eq. (8) for large K, M_n^0 is obtained from Eq. (3) as

$$M_n^0=\frac{1}{2^n}\binom{n}{\frac{n}{2}},\quad n=\text{even} \tag{9}$$

and

$$M_n^0=\frac{1}{2^{n-1}}\binom{n-1}{\frac{n-1}{2}},\quad n=\text{odd}. \tag{10}$$

These values coincide with those of the Hopfield model [2]. This provides evidence of similarity in structures of the mixed states of these two models.

Let us see now if the state is stable. The overlap, after long enough time, reaches an equilibrium value that alone does not provide information about its stability. Hence we turn to a suitable order parameter. According to the dynamics

$$\begin{aligned}S_i(t+1)&=\operatorname{sgn}\left(\frac{1}{\sqrt{N}}\sum_{j\neq i}J_{ij}S_j(t)\right)\\&=S_i(t)\operatorname{sgn}\left(\frac{1}{\sqrt{N}}\sum_{j\neq i}S_i(t)J_{ij}S_j(t)\right),\end{aligned} \tag{11}$$

spin $S_i(t)$ flips if $\Sigma_{j\neq i}S_i(t)J_{ij}S_j(t)<0$, otherwise it stays unchanged. Therefore, the distribution $D(\Lambda)=\langle\delta(\Lambda-\Sigma_{j\neq i}\zeta_i J_{ij}\zeta_j/\sqrt{N})\rangle_\xi$ can be chosen as the parameter to judge the stability of ζ. We find that when $n=2$, D has always a finite value in the region of $\Lambda\leq 0$ for all K. Therefore, there are flips of a fraction of spins and this state ζ is not stable. Figure 2(a) gives the distribution $D(\Lambda)$ for $n=2$ at $K=1$ and 10. Even for very large K (e.g., $K=10$), the distribution is not zero for $\Lambda\leq 0$. This conclusion is also valid for even n. For $n=3$, we observe that the distribution is located at positive Λ, when K is large enough. Figure 2(b) shows the distribution for $n=3$. When $K=1.3$ (corresponding to the storage capacity $\alpha=0.4$), D exists also for $\Lambda\leq 0$; when $K=7$, the distribution resides only in the positive Λ region. The critical values of K (denoted by K_c) are determined numerically and are presented in Table II. Thus ζ is stable when $K>K_c$. This property can be generalized for all odd n, showing a similarity with the Hopfield model [2].

The different behaviors with even n and odd n may be explained by considering the distribution $P_n(\Lambda)$ of Eq. (7), which has maximal value at $\Lambda=0$ for even n when K is large. On the other hand, for odd n the maximum is reached at $\Lambda\neq 0$. With the dynamics, the distribution $P_n(\Lambda)$ evolves to $D(\Lambda)$ indicating a dynamical dependence of $D(\Lambda)$ on $P_n(\Lambda)$. From Eqs. (7),(8) and simulation results we observe that for even n there is a large number of spins (neurons) that maximize $P_n(\Lambda)$ near $\Lambda=0$. It is likely that this leads to nonzero $D(\Lambda)$ around $\Lambda=0$. In the case of odd n, however, there are fewer spins to provide nonzero $P_n(\Lambda)$ near $\Lambda=0$. This is perhaps the reason for zero $D(\Lambda)$ at $\Lambda=0$.

TABLE I. The overlap of ζ with pattern ξ^1 for $n=3$.

K	1	2	3	4	5	10
M_3	0.28	0.35	0.39	0.45	0.48	0.5

TABLE II. The critical values of K versus n.

n	3	5	7
K_c	5.5	8	10

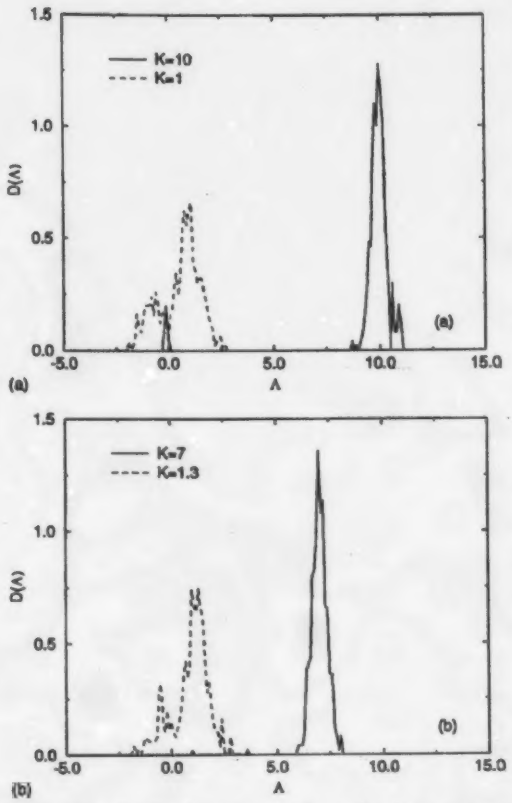

FIG. 2. The distribution D of state ζ for (a) $n=2$ at $K=1$ (dashed line) and $K=10$ (solid line); (b) $n=3$ at $K=1.3$ (dashed line) and $K=7$ (solid line).

The states ζ for odd n have a retrievability (attractor basin), when $K>K_c$. Their behavior can be shown clearly by introducing an overlap, $Q(t)=1/N\Sigma_i^N \zeta_i S_i(t)$, between ζ and a state S that is initially near ζ. Figure 3 gives the evolution of the overlap for $n=3$. After some steps ($s=20$ in Fig. 3) the overlap reaches an equilibrium value for $K>K_c$, while for $K<K_c$ there is no equilibrium value. In many cases of $K<K_c$, we observe two values appearing alternatively after one time step. That means there are two states, ζ_1 and ζ_2, which are reached from the same initial states with a cycle $\zeta_1\rightarrow\zeta_2\rightarrow\zeta_1\rightarrow\cdots$. We point out that both ζ_1 and ζ_2 have the same Hamming distance with any one of the n patterns. The cyclic states are therefore organized in an ultrametrical way [7]: they have the same macroscopical overlaps, but differ in the organization of their components. The states are separated by barriers that are related to the parameter K (the storage limit α). The two branches (dashed and long dashed lines in Fig. 3) of the overlap (corresponding to the two states) at smaller K tend to get closer to the solid curve at larger K. Therefore, for odd n, ζ has a basin of attraction depending on K as in the case of a single state and it forms a metastable state under the dynamics.

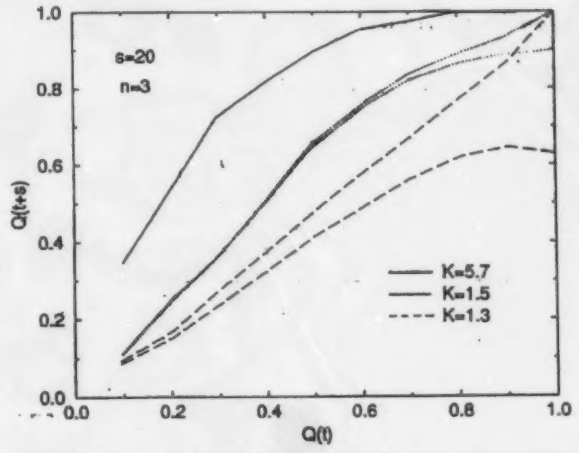

FIG. 3. The evolution of overlap $Q(t)$ for $n=3$ at $K=1.3$ (long dashed lines), $K=1.5$ (dashed lines), and $K=5.7$ (solid line).

The Edwards-Anderson parameter [8,9],

$$R=\lim_{t\rightarrow\infty}\lim_{N\rightarrow\infty}\frac{1}{N}\sum_i^N S_i(t')S_i(t+t') \qquad (12)$$

which compares the configurations at a different time, can also be introduced to study the stability of the states: R is 1 (<1), if the state is stable (unstable). Using R we have obtained the same results as discussed above.

In summary, we have shown both analytically and numerically that mixed states exist in the Gardner model of neural networks. We have also shown that the stability of these mixed states depends on the storage capacity of the model and also on whether n is even or odd. For every odd n, the capacity has a critical value below which the states are stable. In many respects, these states are similar to those of the Hopfield model.

One of the authors (Z.T.) would like to thank L. Schülke and the late B. Schwesinger for useful discussions and their advice. He is also grateful to S. Marculescu for his help. This research was partially supported by a grant to M. K. Ali from the Defence Research Establishment, Suffield.

[1] J. J. Hopfield, Proc. Natl. Acad. Sci. USA 79, 2554 (1982).
[2] D. J. Amit, H. Gutfreund, and H. Sompolinsky, Phys. Rev. A 32, 1007 (1985); Ann. Phys. (New York) 173, 30 (1987).
[3] E. J. Gardner, J. Phys. A 21, 257 (1988).
[4] M. Bouten, A. Engel, A. Komoda, and R. Serneels, J. Phys. A 23, 4643 (1990).
[5] T. B. Kepler and L. F. J. Abbott, J. Phys. (France) 49, 1657 (1988).
[6] Z. Tan and L. Schülke, Int. J. Mod. Phys. B 26, 3549 (1996).
[7] M. Mezard and M. A. Virasoro, J. Phys. (France) 46, 1293 (1985).
[8] S. F. Edward and P. W. Anderson, J. Phys. F 5, 965 (1975).
[9] K. H. Fischer and J. A. Hertz, *Spin Glasses* (Cambridge University Press, Cambridge, 1991).

Reprinted from

PHYSICAL REVIEW
E

STATISTICAL PHYSICS, PLASMAS, FLUIDS, AND RELATED INTERDISCIPLINARY TOPICS

SEPTEMBER 1998
PART B

Pattern recognition in a neural network with chaos

Z. Tan and M. K. Ali
Department of Physics, University of Lethbridge, Lethbridge, Canada T1K 3M4

pp. 3649-3653

Published by
THE AMERICAN PHYSICAL SOCIETY
through the
American Institute of Physics

Volume 58 Third Series Number 3-B

Pattern recognition in a neural network with chaos

Z. Tan and M. K. Ali*
Department of Physics, University of Lethbridge, Lethbridge, Canada T1K 3M4
(Received 17 February 1998; revised manuscript received 6 May 1998)

Chaos is introduced into the Gardner model [J. Phys. A 21, 257 (1988); 22, 1969 (1989)] by reducing the number of connections among neurons. It is shown that patterns can be recognized in this chaotic model by means of chaos control focusing on the history of evolution of the states. Fixed points are not required for pattern recognition in this scheme. [S1063-651X(98)11109-1]

PACS number(s): 87.10.+e, 05.45.+b, 75.10.Nr

Chaos in neural networks has attracted much interest in recent years (see, for example, [1–5]). There are speculations that chaos plays important roles in neural networks. However, a definitive study on the role of chaos in neural networks is still missing. A fully developed chaotic network does not have any disjoint open set in the phase space, meaning that every point can be reached from every other point by running the dynamics. How can such a network perform information processing such as pattern recognition? This is an important and nontrivial question to answer. In this work we address this issue of pattern recognition using a model chaotic network obtained from the well known Gardner model [6].

Tsuda [3] has presented a model for a dynamic link of memory in nonequilibrium neural network. Adachi and Aihara, Nagashima *et al.*, and Nara *et al.* [4] have shown that in the presence of chaos the dynamics wanders among the learned patterns. It is not clear how the correct memory is retrieved from the wandering dynamics. Kushibe *et al.* used the parameter control to carry out recognition by reducing a chaotic model to the Hopfield model [5]. In this paper we demonstrate numerically that the history of trajectories contains information about patterns and we propose an approach to extract this hidden information. Our ansatz is based on synchronization of chaotic systems using the feedback method. In what follows, we first introduce chaos in the Gardner model and then discuss the process of pattern recognition. In the Gardner model [6], N neurons are *fully* connected to one another through the synapses J_{ij} $(i,j=1,\ldots,N)$. The state $S_i(t)$ of the ith neuron at time t is the spin variable $S_i(t)=\pm 1$. In order to store p patterns, $\xi_i^\mu=\pm 1, \mu=1,\ldots,p$, the synapses are trained so that the condition

$$\frac{1}{\sqrt{N}}\xi_i^\mu \sum_{j\neq i} J_{ij}\xi_j^\mu > K \geq 0, \quad \mu=1,\ldots,p, \tag{1}$$

is satisfied. This model can work as an associate memory: It is capable of remembering a maximum of $2N$ patterns and retrieving these patterns, based on partial information about them [6–10]. The patterns coded in synapses are the fixed points under the dynamics of the system. The storage capacity $\alpha=\lim_{N\to\infty} p/N$ is well known [6] to depend on K as

$$\alpha=\left[\int_{-K}^{+\infty}\frac{(t+K)^2}{\sqrt{2\pi}}e^{-t^2/2}dt\right]^{-1},$$

when the net has a maximal (saturation) storage, as to be discussed in this paper. For example, $K=2.00$, 1.00, and 0.68 correspond to $\alpha=0.20$, 0.52, and 0.76, respectively. The limit $\alpha=2$ is reached when $K=0$. The larger the parameter K, the smaller the storage capacity and the larger the basins of fixed points. Details of the Gardner model are available in the literature. In the following, we will discuss the relationship among pattern recognition, storage capacity, and chaotic behavior in the model. Without a loss of generality, we treat the net with maximal storage. Our results show that the ability of recognition depends on K (or storage capacity α), the initial condition, and the number of synapses removed randomly. The net with $N=80$ is used to generate the bulk of our results since this choice gives reliable thermodynamic approximations [10]. We will also discuss below the size effects of the net.

Now we introduce chaos into this model by cutting randomly a number of synapses. Chaos appears if a neuron is connected to fewer than the maximum of $N-1$ neurons in a network of N neurons. The number of missing synapses, chosen randomly, is represented as a fraction c of the total $N-1$ synapses for any neuron. This way of introducing chaos could be significant since a network with partially connected neurons is similar to the human brain in which the neurons are not fully connected [11] and the loss of synaptic connections in the human brain may also occur because of brain damage [2]. In the Gardner model, the cutting off of synapses (CUTS) has been discussed earlier as a diluted approach [12], where the effect of the dilution on the retrieval of patterns corresponding to the fixed points was considered. However, the existing work concentrates exclusively on the concept of pattern recognition being persistent under dynamics without any analysis of its relation to chaos.

We first show that chaos is indeed introduced in this way. The usual way to identify chaos in a system is to calculate the Lyapunov exponents or autocorrelation function [13,14]. We have used both of these techniques to examine the presence of chaos. Here we present only the results of the calculation of autocorrelation function, which is defined as

*Electronic address: ali@hg.uleth.ca

 © 1998 The American Physical Society

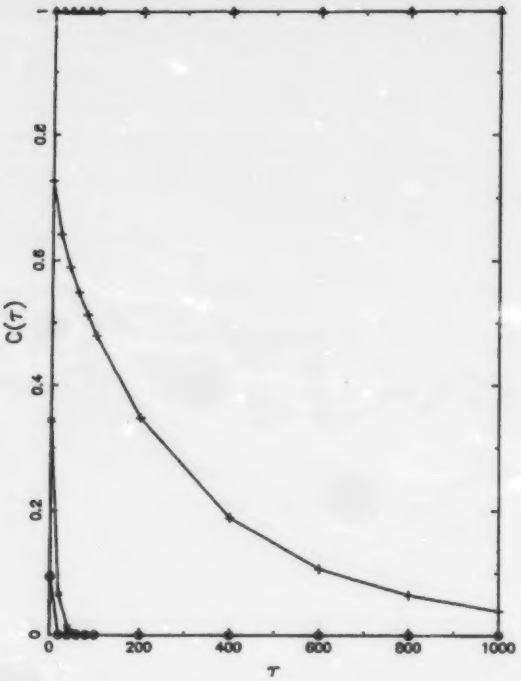

FIG. 1. Autocorrelation function of the Gardner model of $N=80$ at $K=2$ for $c=0$ (triangles), 0.5 (crosses), 0.6 (asterisks), and 0.8 (circles). Points are linked for a better perception.

$$C(\tau)=\lim_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T} S_i(t)S_i(t+\tau), \quad (2$$

where $S_i(t)$ is the state of neuron i at time t. $C(\tau)$ is independent of the choice of i. The criterion of the autocorrelation function states that if $C(\tau)$ is a decreasing function of τ, the system is chaotic. The speed at which $C(\tau)$ decreases is related to the rate at which the systems turns chaotic. The dependence of $C(\tau)$ on τ is presented in Fig. 1 for $N=80$. The figure shows that when $c=0$, $C(\tau)$ remains a constant while for $c>0$, $C(\tau)$ decreases from a finite value to zero. For larger values of c, $C(\tau)$ goes to zero faster, implying a rapid growth of chaotic behavior. It should be pointed out that $C(\tau)$ is also dependent on K. For smaller K (not shown), even a small c will produce chaos.

When chaos is introduced, the network dynamics will not drive an initial state to a fixed point (pattern) as in the nonchaotic model without CUTS. The condition (1) is violated and patterns ξ^{μ} are no longer the fixed points. In fact, a chaotic system has a large number of unstable period orbits [15,16]. Although the dynamics does not drive an initial state to a fixed point, patterns may still be recognized. The synapses are trained before the CUTS and therefore we believe that some information about the patterns may still be present in the remaining J_{ij}. Before a fully chaotic behavior sets in, it may be possible to use the network for pattern identification. We are pointing here to short-term memories. The problem now is to extract this information, if there is any. To this end, we turn to synchronization of identical chaotic systems. The approach to synchronization of chaotic trajectories is not unique [15,16]. Here we utilize the feedback method proposed in [15] and select the trajectory stimulated initially by

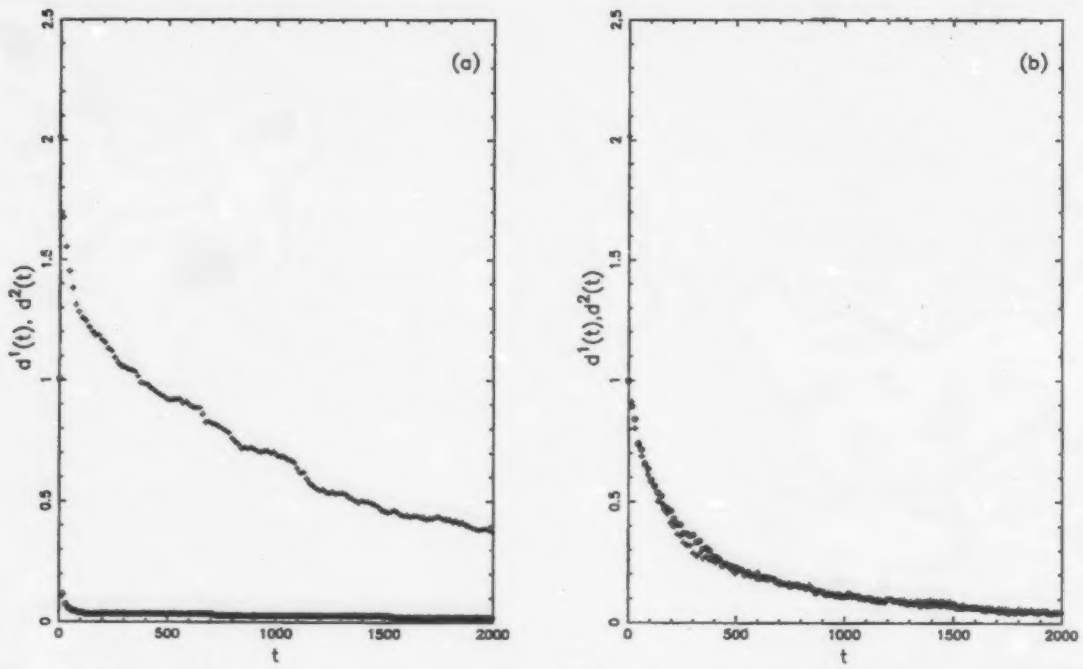

FIG. 2. Dynamic distances $d^1(t)$ (circles) and $d^2(t)$ (crosses) vs time t in the fully connected Gardner model ($c=0$) for $d^1(0)=1$: (a) $K=2$ and $k=0.08$ and (b) $K=0.68$ and $k=0.3$. The pattern can [cannot] be recognized in (a) [(b)].

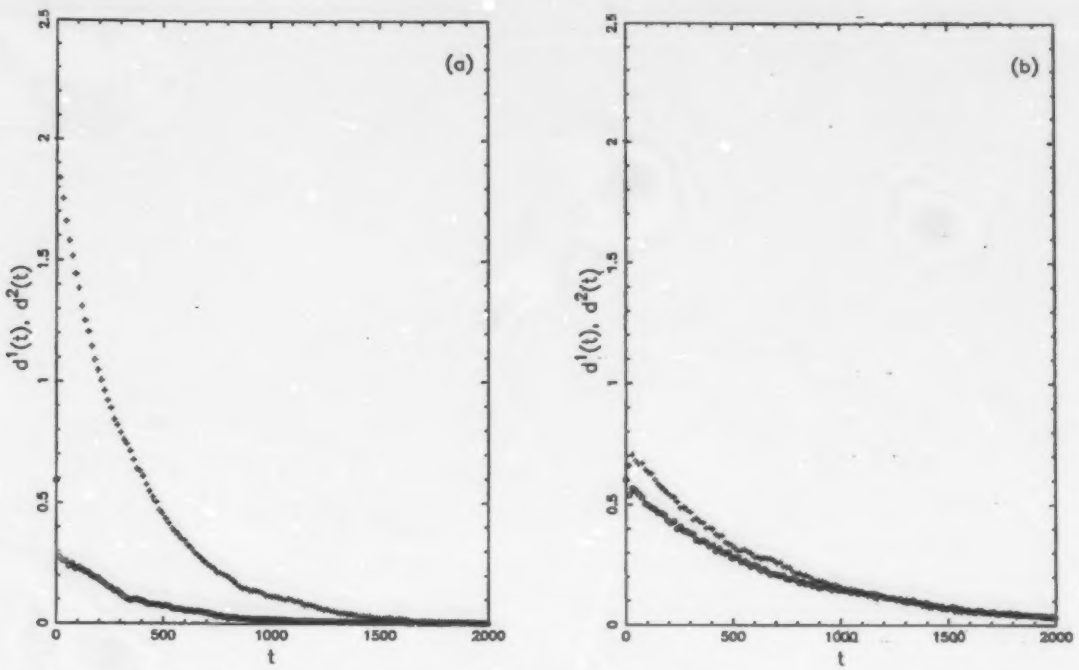

FIG. 3. Dynamic distances $d^1(t)$ (circles) and $d^2(t)$ (crosses) vs time t for $K=2$ and $d^1(0)=0.6$: (a) $c=0.5$ and $k=0.025$ and (b) $c=0.6$ and $k=0.32$. The pattern can [cannot] be recognized in (a) [(b)].

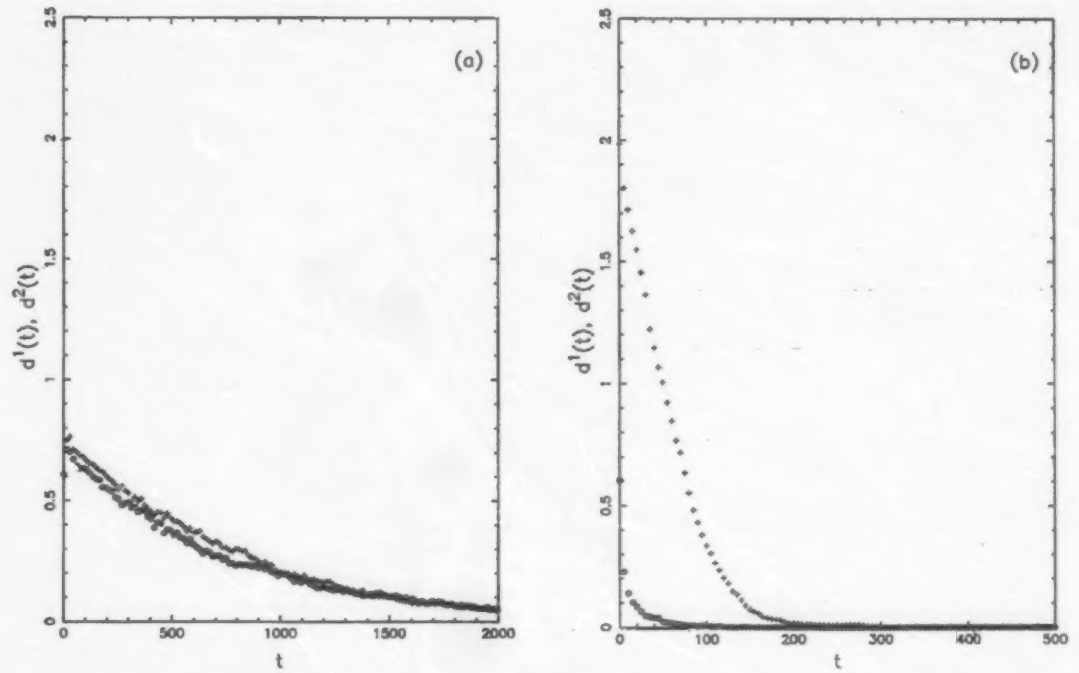

FIG. 4. Dynamic distances $d^1(t)$ (circles) and $d^2(t)$ (crosses) vs time t for (a) $K=1.5$, $c=0.5$, $k=0.4$, and $d^1(0)=0.6$ and (b) $K=2$, $c=0.49$, $k=0.045$, $d^1(0)=0.6$, and $N=200$. The pattern can [cannot] be recognized in (b) [(a)].

TABLE I. Critical values of c vs K for $d^1(0)=0.6$ and $N=80$. The numbers in parentheses are the values of α.

K	c
3.00 (0.1)	0.76
2.50 (0.14)	0.66
2.00 (0.20)	0.50
1.50 (0.31)	0.32

a pattern (e.g., ξ^1) as the desired orbit and synchronize other chaotic trajectories to this orbit. We try to get useful information during the synchronization.

We denote the desired trajectory as $\zeta^1(t)$ with $\zeta^1(0)=\xi^1$ and a trajectory to be synchronized as $s(t)$, which is near ξ^1 at $t=0$. The relation between $\zeta^1(t)$ and $s(t)$ is measured by a dynamic distance

$$d^1(t)=\left\langle \frac{1}{N}\sum_{i=1}^{N}[s_i(t)-\zeta_i^1(t)]^2 \right\rangle_\xi , \tag{3}$$

where $\langle\rangle_\xi$ means the average over random patterns ξ in the Gardner model [6–9]. $d^1(0)$ gives the initial distance. Although $\zeta^1(t)$ is not ξ^1, it is generated by ξ^1 and in a deterministic chaotic system it is unique. Therefore, we "transfer" ξ^1 to $\zeta^1(t)$. If $d^1(t)$ goes to zero (or a small value), ξ^1 is retrieved from $s(0)$ containing only partial information of ξ^1. In this chaotic model, the dynamics will drive $s(t)$ away from $\zeta^1(t)$. $d^1(t>0)$ is larger than $d^1(0)$. However, if at every dynamic step we "help" $s(t)$ to go a little bit closer to $\zeta^1(t)$, we may get useful results. This is just the idea of chaos control by feedback. We now turn to chaos control and try to perform recognition in the chaotic system. The feedback is imposed by modifying the trajectory $s(t)$ at time t with

$$s_i(t)\to s_i^f(t)=s_i(t)+k[\zeta_i^1(t)-s_i(t)], \quad i=1,\ldots,N, \tag{4}$$

where $0<k<1$ is a constant called hereafter the feedback strength. For $k=0$, there is no feedback; for $k=1$, the trajectory is modified immediately to be the desired one. The state is $s^f(t)$ after the feedback process is utilized for the next dynamics. Because the feedback is applied directly and at every time step to the output of the system, $s^f(t)$ will be driven to $\zeta^1(t)$ after a sufficiently long time depending on k. After the synchronization, the feedback term goes to zero and the distance in Eq. (3) vanishes. However, we cannot believe that $d^1(t)=0$ in this case means that $s(t)$ is recognized because $s(t)$ may be synchronized in the same way with all other trajectories $\zeta^2,\ldots,\zeta^p$ coming initially from other patterns, $\xi^2,\ldots,\xi^p$, respectively. Therefore, we need another criterion to distinguish between the synchronized orbits.

The criterion that we have used in this work is based on the "time of synchronization." The phrase time of synchronization (TOS) is used here in a limited sense. The TOS is the time required for the distance between the drive and response systems to become less than a preset value d_0. According to this criterion, the TOS tells us which (target) pattern is retrieved. More precisely, we denote $d^\mu(t)$ as the distance by replacing index 1 in Eq. (3) with μ. The shortest time

FIG. 5. Dynamic distances $d^1(t)$ (circles) and $d^2(t)$ (crosses) started from two random states for $K=2$, $d^1(0)=0.2$, $c=0.5$, and $k=0.023$. The distances are indistinguishable after certain steps.

$$t^r=\min\{t^1,\ldots,t^p\}, \tag{5}$$

where t^μ satisfy

$$d^\mu(t^\mu)<d_0, \quad \mu=1,\ldots,p, \tag{6}$$

determines that pattern r is the target pattern. Here d_0 is a small number. If the shortest time t^r does not exist (the t^μ, $\mu=1,\ldots,p$, are close to one another before d_0 is reached), then the recognition fails. In practice we do not have to compare all distances because after the average in Eq. (3) all patterns are equivalent. Below we set $s(0)$ to be related to ξ^1 with the initial distance $d^1(0)$ and compare the distance $d^1(t)$ between $s(t)$ and $\zeta^1(t)$ and the distance $d^2(t)$ between $s(t)$ and an arbitrary trajectory; see $\zeta^2(t)$ starting from ξ^2. Since the patterns are randomly orthogonal, $d^2(0)$ is 2. In the following, we choose $d_0=0.02$.

The approach has been first checked for the fully connected Gardner model ($c=0$). Figures 2(a) and 2(b) give the distances $d^1(t)$ (circles) and $d^2(t)$ (crosses) versus t with initial distance $d^1(0)=1$ for $K=2$ and 0.68, respectively. In Fig. 2(a) $d^1(t)$ is always smaller than $d^2(t)$ and this leads to the result that t^1 is smaller than t^2, while in Fig. 2(b) $d^1(t)$ and $d^2(t)$ have the same value after certain steps. Therefore, according to our discussion, the network is able to retrieve pattern ξ^1 in Fig. 2(a) but not in Fig. 2(b). This is in agreement with the well-known results obtained by means of fixed points [7–9].

Now we turn to the chaotic model. Figure 3 gives the dynamic distance $d^1(t)$ (circles) and $d^2(t)$ (crosses) versus t for $K=2$, $d^1(0)=0.6$, and different fractions of CUTS and feedback strengths. In Fig. 3(a) with $c=0.5$, $d^1(t)$ goes to zero much faster than $d^2(t)$ and this implies that t^1 is smaller than t^2 when the condition (6) is fulfilled. When $c=0.6$ in Fig. 3(b), we cannot find any difference between $d^1(t)$ and $d^2(t)$ after certain steps. This will not lead to different t^1 and t^2 when the condition (6) is satisfied. Therefore, the pattern is recognized in Fig. 3(a) but not in Fig. 3(b).

It is natural that when c is small, the structure of the fully connected model will not be affected dramatically. Specially, when the fixed points have larger basins (e.g., $K=2$), i.e., smaller storage α, the strong convergent tendency persists. When c is large, things change noticeably. When the basin gets smaller (storage gets larger), even small CUTS may produce chaos.

Figure 4(a) shows the effect of the basin (storage capacity). Compared with Fig. 3(a) for $K=2(\alpha=0.2)$, Fig. 4(a) for $K=1.5$ $(\alpha=0.31)$ does not give distinguishable t^1 and t^2 when the condition (6) is valid. The recognition thus fails in Fig. 4(a). In order to check if there is any size effect, we treated a net with $N=200$ and $K=2$. The results are given in Fig. 4(b). Comparing Fig. 3(a) ($N=80$ and $K=2$) and Fig. 4(b) ($N=200$ and $K=2$), we believe that, for pattern recognition, our results represent large N (thermodynamic) behaviors. We have also checked the nets with $N=40$, 80, and 160 and have found that the results for different sizes are qualitatively the same, except for some quantitative differences. Our results of the effects of different sizes are in agreement with the discussions given in [10].

In short, the ability of recognition depends on the parameters K (which determines the basins of the fully connected nonchaotic model), c (which defines the fraction of CUTS), and $d^1(0)$ (which is the initial distance). Table I gives the critical values of c as a function of K for $d^1(0)=0.6$ and $N=80$. The numbers in parentheses are the corresponding values of α. If the cut ratios c are below the values listed in the table, the recognition is successful; otherwise patterns cannot be recognized. It may be pointed out that the qualitative property is not sensitive to the feedback strength k. k determines the time required for satisfying the condition (6) and does not affect the order in magnitudes among different t^μ.

Using chaos control and the criterion of time of synchronization, we have shown that the information hidden in the history of evolution can be exploited. When chaos exists, the process of pattern recognition can be performed in this way. We emphasize that this conclusion is related not only to the initial condition, but also to the structure of the network that has embedded the patterns into synapses J_{ij}. This can be verified by starting ζ^1 and ζ^2 by two random states instead of ξ^1 and ξ^2. Figure 5 gives the distance $d^1(t)$ between $\zeta^1(t)$ and $s(t)$ and the distance $d^2(t)$ between $\zeta^2(t)$ and $s(t)$. Even for very small initial distance $[d^1(t)=0.2]$ and strong convergent tendency ($K=2$), $d^1(t)$ and $d^2(t)$ quickly become similar and we cannot treat them the way we did in Fig. 3(a). This is not surprising because the initial random states are not stored in the J_{ij}. The network "does not know" the random states. The approach of pattern recognition in chaotic networks proposed in this paper can be easily applied to other chaotic systems. It may be interesting to study the effects of gradual and sudden cutoffs (or regeneration) of synapses on such issues as metastable states [17].

The work has been supported through a grant to M. K. A. from the Defence Research Establishment Suffield under the Contract No. W7702-6-R607/001/EDM.

[1] C. A. Skarda and W. J. Freeman, Behav. Brain Sci. **10**, 161 (1987).
[2] J. Glanz, Science **277**, 1758 (1997).
[3] I. Tsuda, Neural Networks **5**, 313 (1992).
[4] M. Adachi and K. Aihara, Neural Networks **10**, 83 (1997); T. Nagashima, J. Miyazaki, Y. Shiroki, and I. Tokuda, Int. J. Chaos Theory Appl. **2**, 1 (1997); S. Nara, P. Davis, M. Kawachi, and H. Totsuji, Int. J. Bifurcation Chaos Appl. Sci. Eng. **5**, 1205 (1995).
[5] M. Kushibe, Y. Liu, and J. Ohtsubo, Phys. Rev. E **53**, 4502 (1995).
[6] E. Gardner, J. Phys. A **21**, 257 (1988).
[7] E. Gardner, J. Phys. A **22**, 1969 (1989).
[8] T. B. Kepler and L. F. Abbott, J. Phys. (Paris) **49**, 1657 (1988).
[9] Z. Tan and L. Schülke, Int. J. Mod. Phys. B **26**, 3549 (1996).
[10] W. Krauth and M. Mezard, J. Phys. A **20**, L745 (1987).
[11] J. W. Clark, Phys. Rep. **158**, 91 (1988).
[12] M. Bouten, A. Engel, A. Komoda, and R. Serneels, J. Phys. A **23**, 4643 (1990).
[13] A. J. Lichtenberg and M. A. Lieberman, *Regular and Stochastic Motion* (Springer-Verlag, Berlin, 1983).
[14] P. W. Milonni, M.-L. Shih, and J. R. Ackerhalt, *Chaos in Laser-Matter Interaction* (World Scientific, Singapore, 1987).
[15] K. Pyragas, Phys. Lett. A **170**, 421 (1992).
[16] M. K. Ali and J.-Q. Fang, Phys. Rev. E **55**, 5285 (1997).
[17] Z. Tan and M. K. Ali, Phys. Rev. E **57** R3739 (1998).

Discrete Dynamics in Nature and Society, Vol. 2, pp. 243–247
Reprints available directly from the publisher
Photocopying permitted by license only

© 1998 OPA (Overseas Publishers Association) N.V.
Published by license under
the Gordon and Breach Science
Publishers imprint.
Printed in Malaysia.

Pattern Recognition Using Chaotic Neural Networks

Z. TAN, B.S. HEPBURN, C. TUCKER and M.K. ALI*

Department of Physics, University of Lethbridge, 4401 University Drive, Lethbridge, Alberta, Canada T1K 3M4

(Received 10 April 1998)

Pattern recognition by chaotic neural networks is studied using a hyperchaotic neural network as model. Virtual basins of attraction are introduced around unstable periodic orbits which are then used as patterns. Search for periodic orbits in dynamical systems is treated as a process of pattern recognition. The role of synapses on patterns in chaotic networks is discussed. It is shown that distorted states having only limited information of the patterns are successfully recognized.

Keywords: Pattern recognition, Neural network, Chaos

PACS: 87.10.+e, 05.45.+b

I. INTRODUCTION

Recently, studies of chaos in neural networks have drawn attention from workers in diverse fields [1–4]. There are speculations that chaos plays an important role in processes such as pattern recognition in neural networks. However, a definitive study on the role of chaos in neural networks is still missing. The aim of this note is to present a plausible scheme by which chaotic networks can be used in pattern recognition.

First, we recall the following basic features of chaotic systems: (1) In a chaotic system many chaotic attractors may coexist. The basins of these attractors may be riddled. (2) A chaotic attractor is topologically transitive, meaning that every point in the state space can be reached from every other point through dynamical evolution. The frequency at which a randomly initiated trajectory visits the regions of an attractor depends on the dynamical properties of all of the embedded unstable periodic orbits. (3) Periodic orbits are dense in chaotic systems. (4) Chaotic systems are sensitive to initial conditions. This sensitivity implies that chaotic systems lose memory rapidly.

Pattern recognition using chaotic neural networks with such features appears counter-intuitive from the point of view of the usual pattern recognition methods in neural networks. The usual way of pattern recognition in neural networks involves two main steps: (a) the network is taught a certain number of patterns (this determines the synapses) and (b) unknown states which contain partial information of the learned patterns are

* Corresponding author.

recognized as the fixed points reached by running the dynamics. In hard chaos, there are no stable fixed points with attracting basins. Hence, dynamics as such will not drive an arbitrary state to any unstable fixed point as it does in the usual neural network pattern recognition. The fact that chaotic systems have a huge collection of periodic points, each of which may be used as stored patterns, makes chaotic neural networks worth studying. However, one has to be careful. It is possible that the dense periodic points of a chaotic system are clustered in a small region creating practical difficulties to use them. The riddled basin boundaries may also cause problems. We focus on the class of systems in which there is a large number of well distributed periodic orbits, specially those of short periods. Pattern recognition in such chaotic systems can be realized if we augment the dynamics by a scheme that seeks the periodic points and vanishes after the periodic points are found. The procedure (see below), is meant to find the periodic points but not to alter their number, location or nature.

II. THE MODEL DYNAMICAL SYSTEM

For the purpose of demonstration, we consider the "maximum hyperchaotic system" which has recently been studied in Ref. [5]. This model consists of N neurons whose states are described by $S_i(t)$, $i=1,\ldots,N$. The N neurons are connected to one another through the synapses $J_{ij}, i,j=1,\ldots,N$. The dynamics of the states is described by

$$S_i(t+1)=f\left(\sum_{j=1}^{N} J_{ij}S_j(t)\right), \quad i=1,\ldots,N, \quad (1)$$

with

$$f(z)=\tanh(\alpha z)e^{-\beta z^2}, \quad (2)$$

where α and β are constants. Depending on J_{ij} and the parameters α and β, the system presents chaotic or nonchaotic behaviors. For example, when $N=4$, $\alpha=3$ and $\beta=2$, the system is hyperchaotic which can be verified by checking the Lyapunov exponents in the phase space. For the synapses

$$J=\begin{pmatrix} 0.01 & 1.50 & 0.03 & 0.01 \\ 1.60 & 0.01 & 0.10 & 1.00 \\ 0.03 & 0.02 & 0.01 & 1.50 \\ 0.00 & 1.00 & 1.70 & 0.01 \end{pmatrix}, \quad (3)$$

all the four Lyapunov exponents of this 4-dimensional system are positive [5]. The neural network is thus "maximum hyperchaotic".

This hyperchaotic system with $N=4, \alpha=3, \beta=2$ has infinitely many unstable periodic points embedded in a finite phase space. However, only the ones with short periods are better suited for pattern recognition. For pattern recognition, those chaotic systems that support a large number of well distributed periodic orbits of short periods should be preferable. The model system of this work is not claimed to be ideal for this purpose. We have chosen this model to show that the process of pattern recognition by a chaotic system is not affected by its number of positive Lyapunov exponents. Since there are no attracting periodic points, it is unlikely that randomly chosen initial state variables will exactly match those of a periodic point for the dynamics to end in a persistent periodic behavior. Thus for pattern recognition, we need to create "virtual basins of attraction" of the periodic points by augmenting the dynamics in a way that does not change the number, location or nature of the periodic points. This is easily done by an algorithm that finds the periodic point that lies in the nearest vicinity of a given initial point. All the states in this virtual basin of attraction will be identified as the pattern represented by this periodic point. If a periodic point cannot be reached from an initial state, then that state will be categorized as "I do not know". There are algorithms in the literature for finding periodic points of dynamical system. For the purpose of illustration, we use here the classic Newton's root finding method (NRFM) [6]. Using NRFM with $N=4$, $\alpha=3$, $\beta=2$, we have found

TABLE I The fixed points

i	$S_1^{F[i]}$	$S_2^{F[i]}$	$S_3^{F[i]}$	$S_4^{F[i]}$
1	−0.5186	−0.1289	−0.5190	−0.1282
2	−0.1800	−0.6076	−0.3302	−0.6468E−1
3	−0.2428	−0.5491	−0.3536	−0.7057E−1
4	−0.5955	−0.1952	0.1227	0.4119E−1
5	0.2386	0.5679	−0.3522	−0.9470E−1
6	0.1551	0.4656E−1	−0.5736	−0.1758
7	−0.32983	−0.6865E−1	−0.1763	−0.6110
8	0.0	0.0	0.0	0.0
9	0.4224	0.4342	−0.3641	−0.4786
10	0.4005	−0.1048	0.2215	0.5846
11	−0.3158	−0.8375E−1	0.4144	0.4367
12	−0.4756	−0.3918	0.2437	0.6919E−1
13	−0.3575	−0.7517E−1	−0.2542	−0.5388.

25 fixed points of Eq. (1) (roots of $S_i - f(\sum_{j=1}^{N} J_{ij} S_j(t)) = 0, i = 1, \ldots, 4$). These fixed points are denoted by the vectors $\mathbf{Y}_i = (S_1^{F[i]}, S_2^{F[i]}, S_3^{F[i]}, S_4^{F[i]})$ where $S_j^{F[i]}$, $j = 1, 2, 3, 4$ are the four fixed-point values of the state variables for the ith fixed point. Table I contains 13 of these fixed points correct to four significant figures. All the fixed points were generated by a double precision routine. Twelve other fixed points are found by changing the signs of these 12 fixed points (except $\mathbf{Y}_8$). These fixed points are used below for pattern recognition. We have also found a large number of periodic points of higher periods but they are not discussed in this work.

III. PATTERN RECOGNITION

The phase space, in the presence of NRFM, is visualized as disjoint regions of fixed points with virtual basins of attraction. When the dynamics is run in this space with an arbitrary initial state, one of the three things happens: (a) the state evolves into a known fixed point and the pattern is recognized; (b) no fixed point is reached (this is the "I do not know" behavior of the network); (c) a new fixed point different from the known ones is found. This new fixed point is added to the memory. In this way all the fixed points that are accessible by the algorithm can be found if the initial list of fixed points is not complete. One may say that the phase space has been partitioned into regions of dynamic persistence (fixed points with virtual basins).

We now discuss the pattern recognition by considering the fixed points as the patterns themselves. Our task is to retrieve the fixed points. The initial state, e.g. near $\mathbf{Y}_2$ (denoted by $\mathbf{y}_2 = (y_1^2, y_2^2, y_3^2, y_4^2)$), is defined by

$$\mathbf{y}_2 = \mathbf{Y}_2 + \mathbf{e}(R)r, \tag{4}$$

where, $\mathbf{e}(R)$ is a unit vector whose direction varies according to a random number R and r is a real number. The second term in (4) thus gives the distortion of the fixed point $\mathbf{Y}_2$ in the form of a 4-dimensional hypersphere with radius r. Figure 1 illustrates statistically the retrieval rates for the fixed points at various radii. The numbers on the concentric circles are the ratio (percent) of number of successful recognitions to the total probes for recognizing the fixed points. First, the retrievability depends on the noise (radius r) determining how far $\mathbf{y}_2$ is from $\mathbf{Y}_2$. Then, it depends on the patterns. For example, the retrievability for $r = 0.2$ is as high as 0.99% in Fig. 1(a), while it is only 81% in Fig. 1(c). This is a reflection of the location of the fixed points and their virtual basins generated by the NRFM. It may be reminded that the synaptic connection matrix of our model is asymmetric. We would like to stress that it is the root finding method (in this paper NRFM) that determines the fixed points and retrieves a fixed point from a state which has only partial information on it. Without using this approach, the fixed points cannot be found, although we know that they exist.

The scheme presented above for pattern recognition in chaotic systems is different from that used in nonchaotic models. In the usual nonchaotic models patterns are stored in synapses (e.g. [7–9]) such that the dynamics alone drives an initial state having partial knowledge of a pattern to a fixed point (pattern). For chaotic networks, we need an additional algorithm such as the NRFM to drive a state to a fixed point (pattern). Thus, with the help of schemes such as the NRFM, chaotic systems can

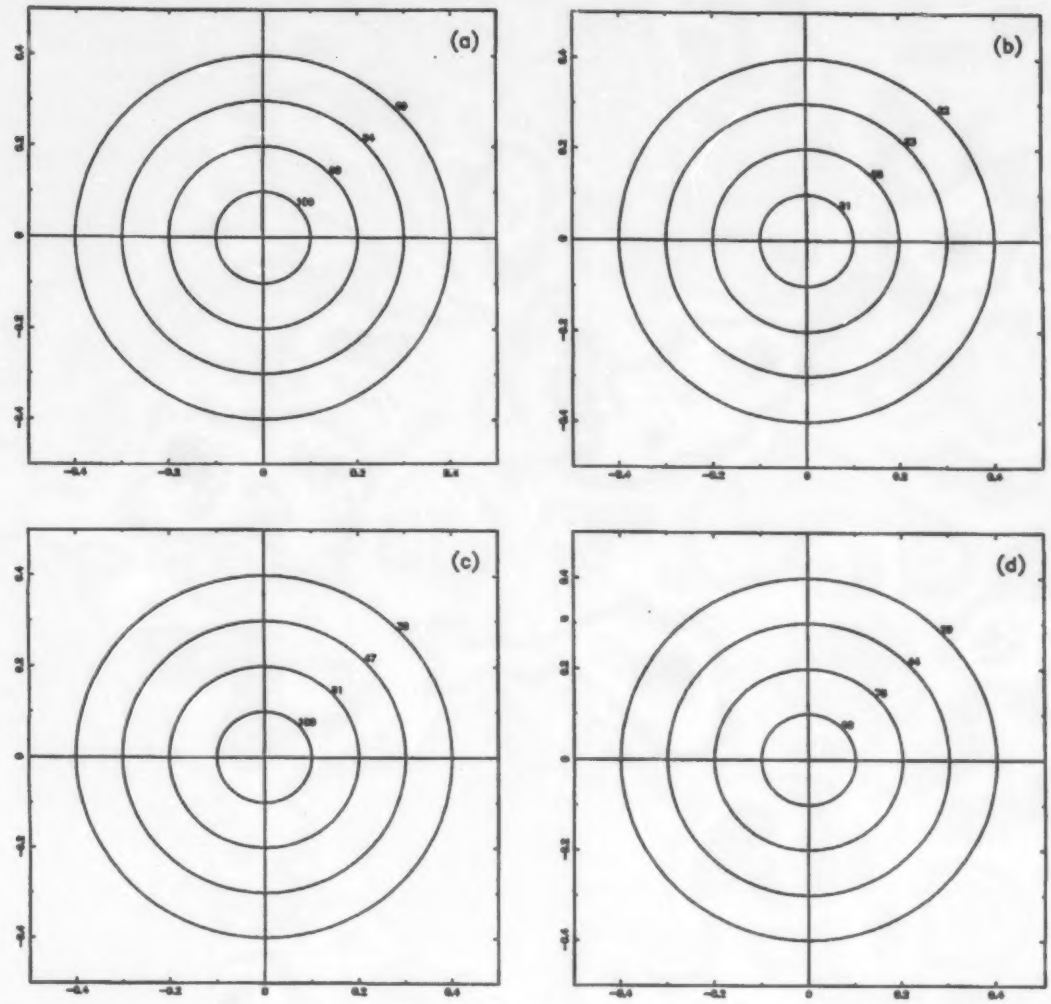

FIGURE 1 The retrieval of the fixed point (a) Y_1; (b) Y_2; (c) Y_5; (d) Y_{10}. The radii of the circles from inner to outer are 0.1, 0.2, 0.3 and 0.4.

be used for pattern recognition. Compared to non-chaotic systems, the role of synapses on patterns is less direct. At this stage, we can only say that the synapses (along with other parameters) influence the number and locations of the distinct fixed points which we have considered as patterns. Search for fixed points (periodic orbits) in dynamical systems is treated in this work as a process of pattern recognition. One advantage of using chaotic systems for pattern recognition is that the storage capacity can be increased enormously. With suitable higher dimensional chaotic systems pattern recognition using chaotic systems may be very useful for practical purposes. In this paper, we have categorized the fixed points of our model system as patterns. The question of mapping a given geometric shape into the fixed points remains to be addressed.

Acknowledgment

This research has been supported by a grant from Defence Research Establishment Suffield under the contract No. W7702-6-R607/001/EDM.

References

[1] Skarda, C.A. and Freeman, W.J., *Behavior and Brain Science*, **10**, 161 (1987).
[2] Glanz, J., *Science*, **277**, 1758 (1997).
[3] Tsuda, I., *Neural Networks*, **5**, 313 (1992).
[4] Nara, S. and Davis, P., *Prog. Theor. Phys.*, **88**, 845 (1992).
[5] Shuai, J.W., Chen, Z.X., Liu, R.T. and Wu, B.X., *Phys. Rev. E*, **56**, 890 (1997).
[6] Press, W.H., Flannery, B.P., Teukolsky, S.A. and Vetterling, W.A., Chapter 9, *Numerical Recipes*, Cambridge University Press (1989).
[7] Hopfield, J.J., *Proc. Natl. Acad. Sci. USA*, **79**, 2554 (1982).
[8] Gardner, E, *J. Phys. A*, **21**, 257 (1988).
[9] Tan, Z. and Ali, M.K., *Phys. Rev. E* (in press).

PATTERN RECOGNITION THROUGH PHASE SYNCHRONIZATION

B.S. HEPBURN AND M. K. ALI

Department of Physics, University of Lethbridge, Lethbridge, Canada, T1K 3M4

It is shown that general synchronization, rather than the complete synchronization, of chaotic lattices may be used for information processing. Pattern recognition is carried out through the phase synchronization of chaotic lattices. The lattices utilized are built by coupling a large set of Lorenz attractors. One lattice is established for each pattern with the coupling determined according to a Hebbian rule. Another lattice is similarly constructed for the unknown pattern. Dynamics are carried out, *from arbitrary initial conditions*, and synchronization is employed between the unknown pattern lattice and each of the known pattern lattices. It is demonstrated that the time required for synchronization of phase can be used as an indication of pattern similarity, without requiring that the dynamics of each lattice become completely synchronized, and maintaining the full chaoticity of the lattice dynamics.

1 Introduction

The standard approaches to pattern recognition with neural networks nearly all rely on fixed point attractors. Memories are encoded in the network by determining the connections between the network elements such that the minimum energy states of the network correspond to memory or pattern states. Around these minima exist basins of attraction — for each attractor there will be a region of network states which all converge on that attractor under normal network dynamics. You can think of the phase space as mapping an energy landscape with hills and valleys and the state of the network as a ball rolling on top of that landscape — the ball always rolls to the bottom of one of the valleys, which represents recalling that memory state. The trick with neural networks is to construct them such that the landscape has a shape such that the right relationship obtains between where the ball begins, and where it ends up. Normally, it is simply a pseudo-euclidian distance criterion that we wish to emulate or the dot-product like Hamming distance measurement for example. That is, whichever memory pattern our input pattern is closest to is the pattern we want to end up with and so we always want the ball to roll to the bottom of the *nearest* valley. This has obvious philosophical problems: 1. if all we are measuring in the end is vector overlap or euclidian distance, why use neural networks and not just dot products say? and 2. given four images such as a rocket, a volkswagen beetle, a skyscraper, and an igloo we

can imagine having good reason to want our network to group the first two together (as vehicles) and the last two (as buildings) rather than grouping long thin objects and squat round ones as would occur with a simple vector comparison.

But rather than abandon neural networks, limitations like these have led scientists to explore the utilization of networks with more complex phase spaces. For instance, the phase space of a chaotic system may be riddled in the sense that any two very near points have diverging trajectories and so initial closeness has little, and in fact nothing in fully chaotic systems, to do with trajectory endpoints. And contemporaneous indications from the neural biological field that chaos might play an essential role in cognition has given rise to many attempts to employ chaos in neural network design. However, researchers remained reluctant to give up their 'fixed-point' point of view. And it would seem rightly so, for it is not at all clear how to utilize chaotic attractors as useful output — they are after all, still chaotic.

Some success has been achieved through control methods, where chaotic behaviour is artificially eliminated as a precondition for recognition, after which the network operates in the usual way as a non-chaotic net [1–4]. Chaos has also been used as a method for escaping spurious energy minima, or as a high-efficiency phase space searching mechanism, but is always, in the end, eliminated upon pattern retrieval. However, methods that rely on the *absence* of chaos do not indicate how chaos itself might be useful. Our contention on the other hand, is that it is better to design chaos *in*, rather than out.

As an example of putting this motto into practice we have been seeking ways to take advantage explicitly of the characteristics unique to chaotic systems. For instance, chaotic systems have a potentially infinite number of unstable attractors and therefore the potential for infinite storage capacity. Of course, one might argue that a machine that recognizes everything in fact recognises nothing, and what's more, infinite storage, even with stable fixed points, would necessitate vanishing basin sizes and therefore the loss of any tolerance for pattern distortion. However this *potentially* infinite capacity in turn allows one complete flexibility in obtaining a compromise between capacity and basin size. This possibility has been explored by us elsewhere [5]. We have demonstrated a technique whereby a root-finding algorithm, such as the Newton-Raphson algorithm, may be used in place of the normal dynamics of a non-linear system. In this way we can access any and as many of the system's natural fixed points (including period-2, period-3, and higher) as we wish. It is important to note that under the normal dynamics these fixed points are unstable, having no basin of attraction at all, and are only truly fixed at infinite precision. With the root finding algorithm however, we are

able to impose *virtual* basins around these points. This is a satisfactory first step in the right direction, but still we are clinging to stability, albeit *virtual* stability. What about no stability whatsoever?

One natural idea to explore is that of synchronization, which might be characterized as *meta*-stability. Our early results in this area are what I wish to discuss today.

2 Pattern Lattices and Phase Synchronization

We have been investigating the use of coupled lattices of chaotic elements for pattern recognition. In these lattices each element, or lattice sight, is a chaotic-oscillator, comprised of its own coupled sub-components. Each sight is then coupled in turn to the other sights, as with a regular neural network. However, our innovation is to use the standard Hebbian rule to determine the coupling strengths between elements of a lattice but to use only *one* of the target patterns per lattice (where the target patterns are the memory set for our network, that is, we want all input patterns to be classified as one of our target patterns). We call the pattern used to determine the connections for a lattice the *parent pattern* for that lattice, and we then create a lattice for each of the target patterns. Each chaotic lattice in other words, has one parent pattern and each of the target patterns is a parent for one lattice. Now, rather than using fixed point behaviour, or even periodic orbits, for pattern retrieval, which would require the network to operate in a non-chaotic regime, we look for similarity between the fully chaotic dynamics of our target pattern lattices and a lattice created for the input pattern. We show that the more similar two patterns are, according to the Euclidian distance between their vectors for example, and therefore the more similar their lattices are, the more similar their dynamics will be.

This is not obvious, however, or trivial. Similarity of dynamics can be as straightforward as total synchronization. But in the area of general synchronization one can find many other measures of similarity. We demonstrate one method of utilizing this dynamical similarity as a measure, namely the synchronization of phase, for pattern recognition. Two non-identical lattices can be phase synchronized without the amplitudes of their state vectors coinciding. The following pair of diagrams illustrate the differences in phase synchronization and total synchronization. In Fig. 1 we see the total amplitudes (the sum of the amplitudes of all lattice components) for 3 slave and master systems, plotted after a certain amount of time under direct feedback. The same slave lattice is used in all three cases but is identical to only one of the masters. We can begin to recognize the phase entrainment of the sig-

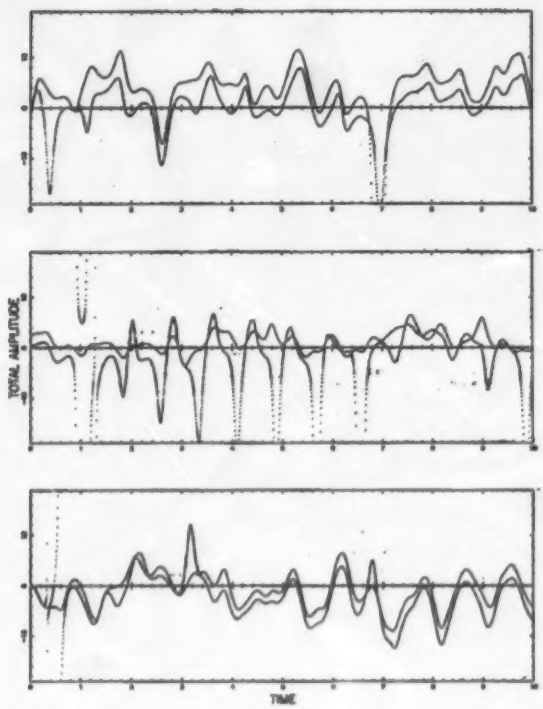

Figure 1. Time signals for master and slave lattices. Plotted is the total amplitude for each lattice formed by summing the amplitude of each component.

nals, and if one were to guess which master the slave is a copy of, one would probably guess the bottom one.

Fig. 2 is for the same time series of data but with the phases calculated, summed, and plotted. We can see that contrary to our guess from the previous plot, it now appears as if the master lattice shown in the top row is the lattice copied for our slave. It is this dynamical similarity that we utilize in our model. Specifically, we show that the *relative* time required for phase synchronization is least between the test pattern lattice and the lattice whose parent pattern is closest to the test pattern. Let us now describe the model.

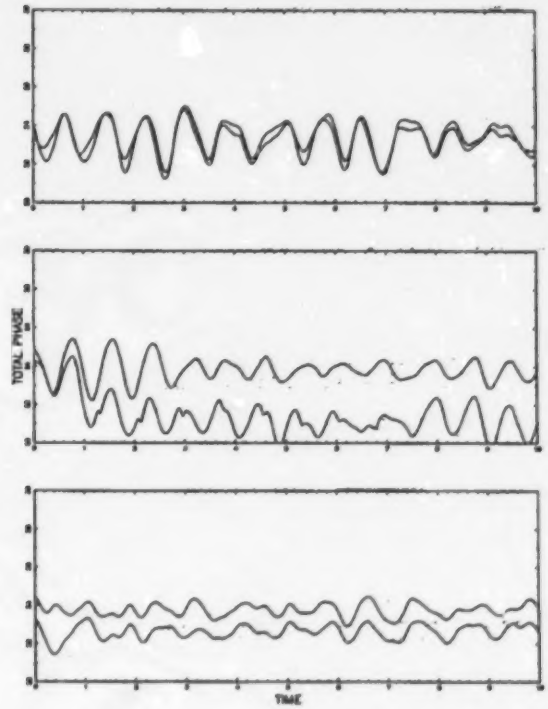

Figure 2. Time signals for master and slave lattices. Plotted is the total phase for each lattice formed by summing the phase of each component.

The Hebbian rule dictates that, for a parent pattern vector $\xi_{1..N}$, the elements of the coupling matrix are

$$W_{ij} = \xi_i \xi_j, \quad i, j = 1, 2, ...N \tag{1}$$

where the ijth element is the coupling strength between the ith and jth lattice sight. For results presented here, only one sub-component of the chaotic oscillator at each sight is coupled. The lattice elements we used are Lorenz oscillators, therefore, with 1st component coupling, the dynamics are governed

by the equations

$$\begin{aligned} \dot{y}_1^i &= 10(y_2^i - y_1^i) + \eta \sum_{j=1}^{N} W_{ij} y_1^j, \\ \dot{y}_2^i &= 80 y_1^i - y_2^i - y_1^i y_3^i, \\ \dot{y}_3^i &= -\frac{8}{3} y_3^i + y_1^i y_2^i \\ & i = 1, 2, ...N \end{aligned} \tag{2}$$

where the number of lattice sights is N, and η is a coupling constant. The subscripts indicate the sub-component of the oscillator, while the superscripts indicate the lattice sight.

The synchronization times were found when patterns were run together two at a time, with the input lattice running as the slave system, and the target pattern lattice as the master, i.e.:

$$Y_{slave} = Y_{slave} + \beta(Y_{master} - Y_{slave}); \tag{3}$$

We have been able to demonstrate, using this method, the feasibility of pattern recognition through *phase synchronization*. For our Lorenz elements, we have obtained encouraging preliminary results using the phase definition introduced by Pikovsky et al.[6]

$$\phi^i = arctan\left(\frac{u^i}{y_3^i}\right) \tag{4}$$

$$u^i = \sqrt{(y_1^i)^2 + (y_2^i)^2} \tag{5}$$

Alternatively, a more generally applicable measurement of phase on the basis of the Hilbert transform

$$\tilde{s}(t) = \pi^{-1} P.V. \int_{-\infty}^{\infty} \frac{s(\tau)}{t - \tau} d\tau \tag{6}$$

would allow one to obtain the phase for any real signal. Our work on this has only recently begun and is ongoing. For purposes here, demonstrating the relative statistical dependence of synchronization times, even with the non-general Pikovsky et al. definition of phase, is enough.

3 Results

Trials were conducted with our model for 10 patterns with coupled lattices of 20 Lorenz attractors, under direct feedback. The dynamics were carried out

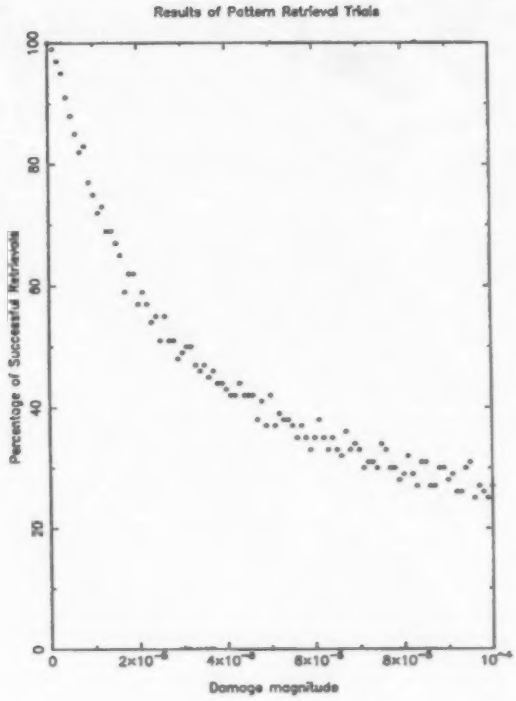

Figure 3. Pattern retrieval rates for coupled Lorenz attractor lattices. Pattern recognition is made on basis of shortest synchronization times

with a fourth-order Runge-Kutta with the feedback added in as a correction after each iteration, and with a coefficient $\beta = .25$. A low coupling strength was used, ($\eta = .05$), which allowed the Lorenz elements to remain in their chaotic domain.

Fig. 3 shows a clear dependence of synchronization times on pattern damage and high retrieval rates for low damages, demonstrating, in principle, the use of relative times as indicators of pattern similarity. However, with standard non-chaotic nets and pattern recognition methods, damages up to ± 0.5 can be achieved with greater than 50 percent success. With our chaotic

nets, recognition drops off much more quickly with increasing damage, with no statistically significant recognitions for damages beyond 0.0009. Hence, our claim is only regarding the demonstration of the principle: similarity of behaviour is indicative, *quantifiably*, of similarity of structure. It remains to be seen how best to use this principle for pattern recognition.

4 Conclusion

There are obvious limitations to our approach. Many lattices running in parallel may not be (currently) feasible as a pattern recognition paradigm, particularly where large numbers of patterns are concerned, and neither does our approach seem to take explicit advantage of the chaos present. However, as a demonstration of a *dynamic* criterion of pattern similarity as opposed to fixed point recognition, and as it does not require the elimination of chaos, this approach has theoretical merit. Namely, the similarity of the patterns, and hence of the lattices, results in a similarity of the dynamics. Chaotic lattices offer incredibly rich behaviour to work with, while phase synchronization allows us to measure the similarity of that behaviour dynamically.

References

1. Skarda, C.A. and Freeman, W.J., *Behavior and Brain Science*, **10**, 161 (1987).
2. Glanz, J., *Science*, **277**, 1758 (1997)
3. Tsuda, I., *Neural Networks*, **5**, 313 (1992)
4. Nara, S. and Davis, P., *Prog. Theor. Phys.*, **88**, 845 (1992).
5. Z. Tan, B.S. Hepburn, C. Tucker, and M. K. Ali, DDNS, **2**, 243 (1998).
6. Pikovsky, A.S., et al., Physica D, **104**, 219 (1997).